Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Escuela Profesional de Ingeniería de Software

DoLaw: buscador semántico especializado para la legislación peruana de tecnologías de información

TESIS

Para optar el Título Profesional de Ingeniero de Software

AUTOR

Diego Augusto OTOYA PAZ

ASESOR

David Santos MAURICIO SANCHEZ

Lima, Perú

2019

Referencia bibliográfica

Otoya, D. (2019). *DoLaw: buscador semántico especializado para la legislación peruana de tecnologías de información*. Tesis para optar el título profesional de Ingeniero de Software. Escuela Profesional de Ingeniería de Software, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

HOJA DE METADATOS COMPLEMENTARIOS

CÓDIGO ORCID DEL AUTOR: 0000-0003-3209-0170

CÓDIGO ORCID DEL ASESOR: 0000-0001-9262-626X

DNI DEL AUTOR: 47923874

GRUPO DE INVESTIGACIÓN: Facultad de Ingeniería de Sistemas e Informática

INSTITUCIÓN QUE FINANCIA PARCIAL O TOTALMENTE LA INVESTIGACIÓN:

Universidad Nacional Mayor de San Marcos

UBICACIÓN GEOGRÁFICA DONDE SE DESARROLLÓ LA INVESTIGACIÓN, DEBE INCLUIR LOCALIDADES Y COORDENADAS GEOGRÁFICAS:

Cercado de Lima, LIMA, PERÚ

Coordenadas: -12.053427, -77.085709

AÑO O RANGO DE AÑOS QUE LA INVESTIGACIÓN ABARCÓ:

2018-2019

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, DECANA DE AMÉRICA
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE

Acta de Sustentación de Tesis

Siendo las 4pm del día 20 de mayo del año 2019, se reunieron los docentes designados como miembros de Jurado de la Tesis, presidido por la Dra. Nora Bertha La Serna Palomino, Lic. Jorge Luis Chávez Soto (Miembro), y el Dr. David Santos Mauricio Sánchez (Miembro Asesor) para la sustentación de la Tesis intitulada: "DOLAW: BUSCADOR SEMÁNTICO ESPECIALIZADO PARA LA LEGISLACIÓN PERUANA DE TECNOLOGÍAS DE INFORMACIÓN"; por el Bach. **Diego Augusto Otoya Paz**, para optar el Título Profesional de Ingeniero de Software.

Acto seguido de la exposición de la Tesis, el Presidente invitó al Bachiller a dar respuesta a las preguntas establecidas por los Miembros de Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros de Jurado, el bachiller obtuvo la nota de 18.. (En letras)... DIECIOCHO

A continuación el Presidente del Jurado, Dra. Nora Bertha La Serna Palomino declara al Bachiller **Ingeniero de Software**.

Siendo las .... horas, se levantó la sesión.

.....
Dra. Nora Bertha La Serna Palomino
Presidente

.....
Lic. Jorge Luis Chávez Soto
Miembro

.....
Dr. David Santos Maurico Sánchez
Miembro Asesor

A mis padres, Luis y Judith, por ser ejemplo para mi vida, por enseñarme a decidir lo mejor para mi presente y mi futuro, por no parar de darme consejos y estar al servicio de mí y mis objetivos, por poner su corazón en sus hijos, mi hermana y yo, a pesar de las adversidades.

A mi hermana, mis abuelas, mi novia y al resto de mi familia, cada quien a su manera, por impulsarme día a día a ser responsable, a seguir adelante a pesar de los problemas, así como ellos lo han hecho, a ponerme retos y demostrar que puedo superarlos.

AGRADECIMIENTOS

Al asesor, Dr. David Mauricio Sánchez, por el apoyo y el tiempo brindado para el desarrollo de esta tesis.

A los miembros del jurado, Jorge Chavez Soto y Nora La Serna Palomino, por prestar su evaluación crítica y desinteresada en el proyecto realizado.

A la Universidad Nacional Mayor de San Marcos y la Facultad de Ingeniería de Sistemas e Informática, por haber invertido en mi educación.

A mis padres, por su apoyo incondicional y por el empuje que admiro en ellos.

A mi hermana, por darme consejos basados en su experiencia profesional.

A mi novia, por demostrarme que la vida necesita un toque de amor para que al cumplir tus objetivos seas más feliz.

A todas aquellas personas que indirectamente me ayudaron para culminar este trabajo y que muchas veces constituyen un invalorable apoyo.

DOLAW: BUSCADOR SEMÁNTICO ESPECIALIZADO PARA LA LEGISLACIÓN PERUANA DE TECNOLOGÍAS DE INFORMACIÓN

RESUMEN

El presente trabajo de investigación tiene por objetivo el desarrollo de un programa especializado en la búsqueda de documentos en legislación peruana de tecnología de información, a través de la interpretación semántica de las palabras clave que el usuario final introduce. La finalidad es realizar una búsqueda en el contenido completo de dichos documentos, con elementos funcionales personalizados para la legislación peruana de tecnología de información, diseñados para facilitar la búsqueda al brindar funcionalidades adicionales específicas para dicha legislación.

Durante el proyecto se definen distintos procesos: análisis de las consultas del usuario, análisis del contenido de la legislación, indexación, generación de consultas ponderadas por prioridad, ordenamiento de resultados obtenidos, entre otros, con el fin de satisfacer las necesidades de los usuarios, sin afectar la precisión y exhaustividad de los resultados.

Palabras clave: buscador, documentos, búsqueda semántica, interpretación, precisión, exhaustividad

DOLAW: SEMANTIC SEARCH ENGINE SPECIALIZED FOR THE TREATMENT OF PERUVIAN INFORMATION TECHNOLOGY LEGISLATION

ABSTRACT

The objective of this research is the development of a search engine specialized on searching documents in Peruvian information technology legislation, through the semantic interpretation of the keywords that the end user introduces, searching on the complete content of those documents, including specialized functional elements useful for peruvian information technology legislation, designed to facilitate the search process by providing additional specific functionalities for the Peruvian legislation.

During the project we defined different processes: Analysis of user queries, analysis of the legislation content, generation of queries weighted by priority, ordering obtained results, and others. All these processes are established to satisfy the user's need, without affecting precision and recall.

Keywords: search engine, information retrieval, documents, semantic, interpretation, precision, recall

ÍNDICE

Lista de Figuras

Lista de Tablas

CAPÍTULO 1: INTRODUCCIÓN

1.1 Antecedentes del Problema

Los seres humanos siempre han buscado satisfacer sus necesidades de información utilizando cualquier medio, siendo el medio que más perdura aquel que se da a través de la comunicación escrita. Esta se ha dado desde los primeros manuscritos, pasando por el uso de la imprenta, hasta llegar a los distintos textos digitalizados que abundan en internet. De hecho, la búsqueda de información a través de un computador empezó en los años 40 (Liddy, 2005), y hoy que se vive en la sociedad de la información, se busca aprender cada vez más, por lo que la cantidad de fuentes de información ha crecido de forma exponencial, a tal grado que ya no solo existen textos informativos, sino que es fácil toparse con información meramente distractora o incluso con desinformación o falsedad.

Entonces, ¿cómo lidiar con la información que no interesa? La tecnología avanza muy rápido, dirigiéndose siempre a solucionar problemas y optimizar procesos. Es por ello que no es admisible que se pierda tiempo verificando cuál es documento de un conjunto que tiene lo que se quiere. Por suerte, es la propia tecnología la que facilitó la herramienta definitiva para solucionar este problema: los buscadores de documentos, los cuales son la forma dominante de acceder a información (Manning, Raghavan, & Schütze, 2008). Estos buscadores consisten en leer un conjunto de palabras ingresadas por el usuario, y ubicar los documentos que contengan referencias a estas frases dentro del contenido de los textos de un banco de infomación.

Pero, en cuanto a los buscadores de documentos, hay que mencionar que existen buscadores genéricos que funcionan correctamente. Estos solicitan cargar una carpeta donde se encuentren todos los documentos en los cuales buscar, para que, al terminar, se pueda encontrar coincidencias

exactas con la palabra que se ingresa. Entonces, ¿por qué hacer una investigación y desarrollar un nuevo sistema si el problema parece ya tener solución? Porque necesitamos profundizar en la capacidad del buscador y configurarlo de manera especializada. Por ejemplo, no podemos limitarnos a encontrar únicamente coincidencias exactas con la frase de búsqueda, ya que debemos considerar que usualmente el usuario ingresa mal el texto a buscar, o la consulta tiene problemas ortográficos, o el texto en el que busca tiene un problema ortográfico, o inclusive el documento tiene el contenido que el usuario busca, pero está expresado de otra manera (por ejemplo, con sinónimos).

Entonces, hay mucho por mejorar en los sistemas que hacen búsqueda por coincidencia exacta. Ahora, si se enfoca el desarrollo con una orientación hacia una temática determinada, se puede, por ejemplo, identificar la orientación semántica de las consultas de los usuarios, o establecer filtros especializados para dicha temática, o agregar otras funcionalidades pertinentes que permitan dar valor agregado al producto. Definitivamente, hacen falta buscadores específicos (Hanauer, Mei, Law, Khanna, & Zheng, 2015), porque los buscadores de dominio específico agregan valor al explotar los conocimientos de sus respectivos dominios (Schmidt, Schnitzer, & Rensing, 2016).

Por otro lado, existen diversos algoritmos necesarios para construir un buscador; entre ellos, algoritmos para el posicionamiento de los resultados, y para la sugerencia de resultados similares o para la corrección ortográfica. Además, el problema puede ser abordado de muchas maneras, y la manera a elegir va a depender del enfoque y sector al que se quiera aplicar el buscador, ya que algunas cosas que son muy necesarias para un escenario pueden ser descartables para otro. Esa es la magia de un buscador personalizado para un dominio específico.

Actualmente, existen muchas áreas que no cuentan con un buscador personalizado (Hanauer, Mei, Law, Khanna, & Zheng, 2015). En esta oportunidad, se ha elegido la legislación peruana,

específicamente en tecnología de información (en adelante TI), debido a que esta es un área en crecimiento que genera constantemente nuevos términos y, a su vez, constantemente aparece nueva legislación para regular cada una de las funcionalidades que la tecnología permite y añade día a día (Ferreyros, 2016).

1.2 Definición del Problema

El problema consiste en la búsqueda y ubicación de legislación peruana de TI, del período entre el 2000 y el 2018, a través de la interpretación semántica de las palabras clave que el usuario final introduce, de modo que se pueda realizar una búsqueda en el contenido completo de dicha legislación de lo que el usuario quiere encontrar.

1.3 Importancia del Problema

Existe legislación desde el año 1904 entre leyes, resoluciones legislativas, decretos leyes, leyes regionales expedidas por los Congresos respectivos que creó la Constitución de 1920, decretos legislativos, decretos de urgencia y otras con rango o fuerza similar (Congreso de la República del Perú, 2016). 30 423 están almacenadas en el portal del Congreso de la República en el Archivo Digital de la Nación. De ellas, no se sabe con exactitud cuántas son de TI, porque no se ha tenido un control ni administración sobre la legislación para agruparlas por tópico. Los buscadores de legislación que actualmente se usan (Archivo Digital de la Nación y Sistema Peruano de Información Jurídica) solo diferencian la jerarquía o categoría a la corresponde una ley y la fecha de publicación de la misma; y para los usuarios, dichas funcionalidades terminan siendo muy limitadas. En resumen, los buscadores de legislación peruana se encuentran en este estado de

insuficientes capacidades, e inclusive no existe ningún buscador de legislación peruana especializado en un solo tema, tal como se plantea hacer (con la legislación de TI).

Por otro lado, la carga legal en el Perú es alta. Por ejemplo, en el 2014, el Perú ingresó 1 039 428 juicios, de los cuales 972 661 fueron atendidos, entre las fiscalías supremas, superiores y provinciales (Ministerio Público: Fiscalía de la Nación, 2015); es decir, casi 100 000 juicios quedaron pendientes para el 2015. Son cifras bastante grandes y, a pesar de que se redujo la cantidad de juicios pendientes del 2013 (Ministerio Público: Fiscalía de la Nación, 2014), sigue siendo una gran carga de trabajo que se vería notablemente reducida con la ayuda de un sistema más preciso. Un buscador proporcionaría la información oportuna en el momento oportuno y, así, ayudaría a acelerar la atención a los juicios.

Para continuar hablando con cifras, se investigó que para los juicios de alimentos, en el Perú, se toma en promedio de dos a tres semanas para admitir o no admitir una demanda (dependiendo si cumplen los requisitos). Luego, se concreta una audiencia, según la disponibilidad del juez, fecha que normalmente se fija entre 2 a 3 meses después la fecha de inicio de demanda (Radio Capital: Rosa María Palacios, 2011). Eso demuestra que los jueces tienen alta carga de trabajo; periodo en el cual el propio juez y el personal legal o asistente generan gastos por servicios al Estado. Como es evidente, parte del tiempo que emplean los funcionarios legales lo ocupan en revisar y repasar las leyes, pues en este trabajo se necesita conocer perfectamente la legislación, por lo que, si se les facilita esta tarea, podrían acelerar la velocidad de atención de juicios, y con ello, además, reducir personal y gastos.

1.4 Motivación

En primer lugar, no existe un buscador especializado para legislación peruana sobre tecnologías de información (Ferreyros, 2016). Si bien existen buscadores para leyes peruanas, estos no ofrecen las funcionalidades personalizadas suficientemente útiles para brindarle al usuario una agradable experiencia de búsqueda, ya que o bien solo realizan una búsqueda simple por coincidencia de la frase clave buscada, o bien están basados en metadatas, o no están especializadas en tecnología de información. Se debe desarrollar de una mejor manera un buscador que tenga en cuenta las características específicas de la búsqueda de legislación peruana de tecnología de información.

Además, cada día se crean nuevos proyectos de ley, decretos, órdenes municipales, entre otros (Diario El Peruano, 2019), por lo que es urgente la presencia de una herramienta que soporte la dinamicidad de esta documentación, y en el caso de TI, si bien es un sector joven, tiene una larga vida por delante para crecer, pudiendo tal vez ser el área más importante del futuro, por lo que el desarrollo de una herramienta en esta área es una necesidad inminente.

1.5 Objetivos del Estudio

1.5.1 Objetivo Principal

El objetivo principal de este estudio es desarrollar un buscador especializado para la búsqueda a texto completo de documentos de la legislación peruana de TI en el periodo entre el 2000 y el 2018, a partir de un repositorio, utilizando interpretación semántica de las palabras claves que el usuario introduce, con la finalidad de obtener documentos más relevantes.

1.5.2 Objetivos Secundarios

Durante el desarrollo habrá necesidad de herramientas (llámese algoritmos, conocimientos, librerías, técnicas, entre otras) que ayuden a la implementación de las características planteadas en el objetivo principal, para que así el producto final pueda cumplir con las expectativas planteadas; para ello, en capítulo de aporte se planteará los módulos y funcionalidades que deben estar presentes en el sistema para cumplir dichas expectativas.

Asimismo, para el cumplimiento del objetivo principal, durante el desarrollo del sistema se cumplirá los siguientes objetivos específicos:

- OS1: Revisar el estado de la literatura para conocer las mejores prácticas y tendencias en el desarrollo de buscadores
- OS2: Diseñar una arquitectura para la búsqueda de legislación peruana de TI
- OS3: Desarrollar los componentes de interpretación, indexación, búsqueda y ordenamiento, que forman parte del buscador e integrarlos
- OS4: Validar el buscador realizando encuestas para determinar la satisfacción del usuario en relación la relevancia de los documentos sobre la legislación peruana de TI.

1.6 Propuesta

Proponemos el desarrollo de un buscador semántico especializado para la búsqueda de legislación peruana de TI, el cual sea capaz de interpretar las frases de búsqueda ingresadas por el usuario, para generar búsquedas paralelas sugeridas y así al buscar en el contenido completo de la legislación, encontrar mayor cantidad de resultados positivos, que serán ordenados por prioridad, dando un mayor peso a la coincidencia con la frase de búsqueda original que a las consultas paralelas.

Hay que mencionar que, al ser un buscador personalizado para la legislación peruana de TI, este debe interpretar tecnicismos tanto legales como tecnológicos, debe conocer y traducir siglas comunes internacionales y sobre todo peruanas, debe entender regionalismos, y debe estar parametrizado para buscar por subconjuntos, ya sea por leyes de cierta temática, por fecha de publicación o por jerarquía.

Para ello se debe diseñar una solución que se encargue de procesar la consulta del usuario para así interpretarla y buscar lo que dijo y lo quiso decir (búsqueda semántica) dentro del repositorio de legislación, con las condiciones que el usuario haya consultado, para luego encontrar y obtener la legislación resultante de la consulta, y presentarla al usuario debidamente ordenada por prioridad.

La idea general de la propuesta puede observarse en la figura 1.

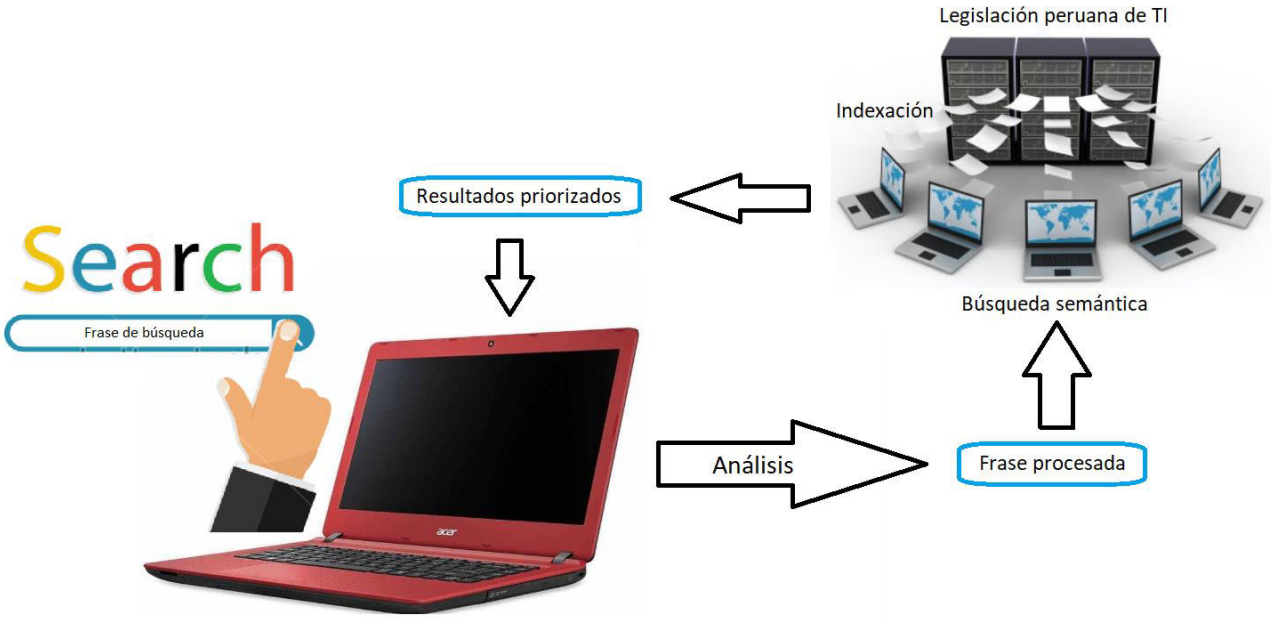


Figura 1. Idea general del buscador propuesto

1.7 Organización de Tesis

Esta tesis está constituida por 5 capítulos, que a continuación se detalla.

En el capítulo 2, se realiza una revisión de la literatura sobre las búsquedas de documentos, donde se plantea una metodología de investigación, planificándola y ejecutándola, para luego analizar los resultados obtenidos.

En el tercer capítulo, se diseña y desarrolla el buscador, así, se hace una breve introducción del porqué se estableció el alcance del buscador, para luego mencionar la metología de desarrollo y el propio desarrollo del sistema.

En el cuarto capítulo, se realiza la validación del sistema terminado, a través de la comparación de la satisfacción de los usuarios de cada uno de los sistemas de búsqueda de legislación.

Por último, en el quinto capítulo, se presenta las conclusiones, los posteriores trabajos futuros que se pueden realizar para ampliar el estudio realizado en esta tesis, así como las limitaciones.

CAPÍTULO 2: MARCO TEÓRICO: LEGISLACIÓN PERUANA

En este capítulo se brinda una vista general sobre la definición y el estado de la legislación peruana, haciendo énfasis en la legislación de tecnología de información.

2.1 Definición y organización

La legislación es un conjunto de leyes establecidas y reguladas por un Estado u otra autoridad perteneciente a la estructura del poder judicial. En el Perú, la legislación peruana actualmente es alojada por el Congreso de la República, en un portal dedicado para este fin llamado El Archivo Digital de la Legislación del Perú, en el cual se almacenan normas con rango de ley que pertenecen al sistema normativo peruano, así como las Leyes de Indias (Congreso de la República del Perú, 2016). El archivo de toda la legislación peruana se encuentra en la web del Congreso de la República, a través del enlace http://www.leyes.congreso.gob.pe/

La legislación peruana actualmente se puede organizar en dos grupos: leyes no numeradas, promulgadas desde el siglo XIX hasta 1904, y leyes numeradas de 1904 a la fecha. Se mantiene el registro del primer grupo a modo de imágenes y meramente con fines archivísticos. Entre la legislación numerada y vigente, existen leyes, resoluciones legislativas, decretos leyes, leyes regionales expedidas por los Congresos respectivos que creó la Constitución de 1920, decretos legislativos, decretos de urgencia y otras con rango o fuerza similar (Congreso de la República del Perú, 2016). Dicha clasificación de la legislación peruana responde a un rango de jerarquía, siendo la Constitución peruana el tipo de ley con la mayor jerarquía en los distintos rangos.

La jerarquía de la legislación peruana se suele representar en una pirámide de Kelsen, representación piramidal didáctica, en donde se colocan los rangos de normas jurídicas por encima de otras, según su predominancia al momento de su aplicación, ubicando la Constitución peruana

en el pico de la pirámide. Una representación de pirámide de Kelsen de la legislación peruana la podemos ver a continuación en la figura 2.

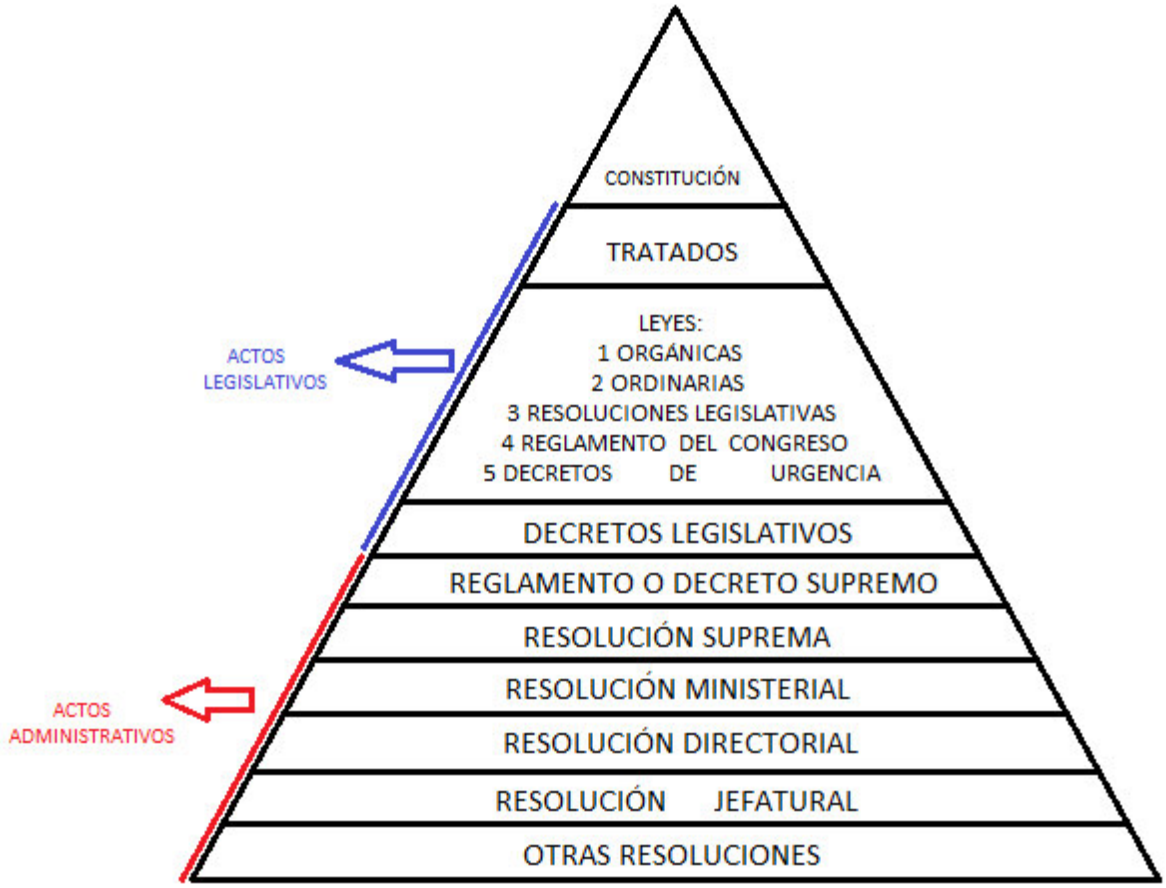


Figura 2. Pirámide de Jerarquía de legislación peruana

Resaltar una vez más que, siguiendo el principio de jerarquía de la norma jurídica, ninguna norma inferior puede mandar sobre una norma superior, pues se tiene que respetar el orden de jerarquía de la pirámide.

2.2 Buscadores de legislación peruana

En este caso de estudio, la palabra "buscador" se refiere a aquellos sistemas que permiten encontrar y obtener documentos de un repositorio a través del contenido de los documentos; documentos que en este caso son legislación de diferente tipo. Entonces, bajo este concepto, oficialmente existen dos buscadores públicos de legislación peruana:

- El Sistema Peruano de Información Jurídica (SPIJ - http://spij.minjus.gob.pe/), propiedad del Ministerio de Justicia, cuya referencia la podemos ver en la figura 3.
- El archivo digital del Congreso de la República (http://www.leyes.congreso.gob.pe/), cuya referencia se encuentra en la figura 4.

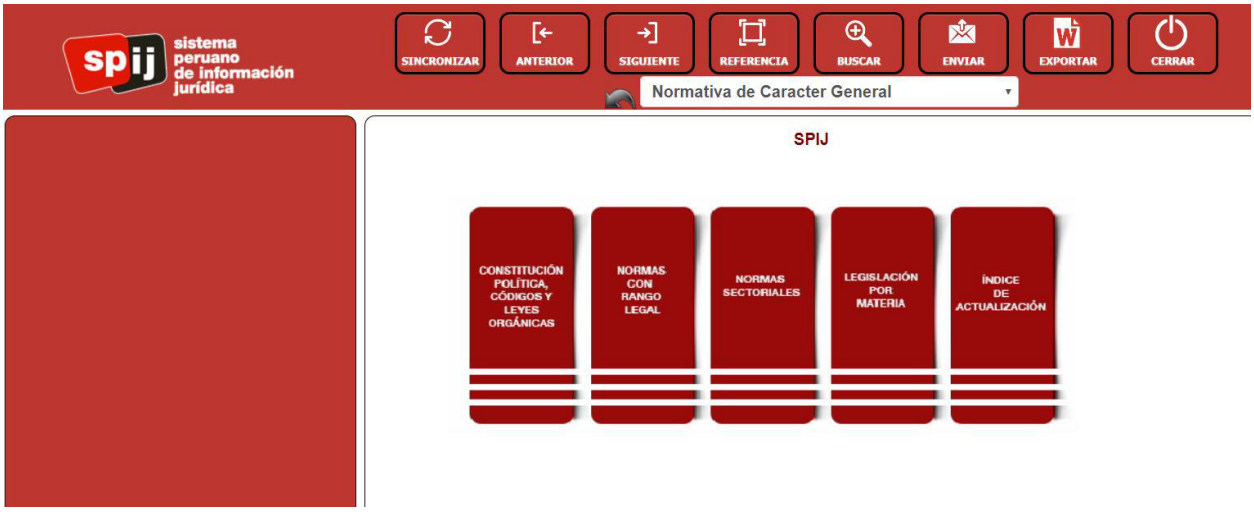


Figura 3. Portal del Sistema Peruano de Información Jurídica

Fuente: http://spij.minjus.gob.pe

Figura 4. Portal del Archivo Digital del Congreso de la República

Fuente: www.leyes.congreso.gob.pe

Se mencionarán las capacidades y limitaciones de estos buscadores en el subcapítulo 4.2.1.1, durante el análisis de los requerimientos, pues allí se definen las características que debería alcanzar o superar el sistema desarrollado en comparación con los existentes.

2.3 Legislación peruana de TI

Se define a la legislación de TI a la legislación que regula bienes o servicios informáticos, así como los datos obtenidos por estos medios, los procesos y el modo de almacenamiento de dichos datos. Actualmente, a finales del 2018, el único sitio web que almacena únicamente legislación peruana de TI es el portal de informática jurídica, http://www.informatica-juridica.com/,

desarrollado por José Cuervo, a modo de proyecto personal abierto al público, pero dicho portal no incluye un buscador que encuentre las palabras clave que ingresa el usuario dentro del contenido de toda la legislación cargada, solo tiene presente un buscador que permite encontrar aquellos documentos cuyo título coincida con las palabras que el usuario ingresa, siendo dicho título uno establecido por el administrador del portal, además de no haber otras características adiciones necesarias como un filtro.

Por ello, el método de búsqueda más fiable para estudiantes y estudios jurídicos para buscar legislación peruana de tecnología de información es usar los buscadores oficiales mencionados en el subcapítulo 2.2, SPIJ y el Buscador del Archivo Digital, ingresando frases de búsqueda con palabras relativas a la tecnología de información, con las limitaciones que conlleva usar estos buscadores (dichas limitaciones se mencionan durante el diseño del sistema propuesto, en el capítulo 4, pues fueron conversadas durante el análisis de requerimientos).

CAPÍTULO 3: REVISIÓN DE LITERATURA SOBRE BÚSQUEDAS DE DOCUMENTOS

En este capítulo se hará una revisión de los artículos sobre la búsqueda de documentos, en particular aquellos que se presenten algoritmos de búsqueda, herramientas para el desarrollo de buscadores y formas de evaluar un buscador. Además, se realizará un análisis de los artículos seleccionados sobre estos temas.

3.1 Metodo de Investigación

Para la revisión de la literatura, se ha considerado tres fases: planificación, desarrollo y resultados. Esta metodología la usa Santisteban & Mauricio (2017), Kitchenham & Charters (2007), entre otros, y se desarrolla siguiendo tres pasos:

- Planificación: Definifición de preguntas de investigación y de un protocolo para la búsqueda de los artículos.

- Ejecución: Se aplica el protocolo definido en la planificación en diferentes repositorios bajo un mismo criterio.

- Resultados: Se presentan los resultados obtenidos al realizar el análisis de los artículos obtenidos.

3.2 Planificación

Con la finalidad de encontrar documentos adecuados acerca de la búsqueda de documentos, se plantean las siguientes preguntas:

Q1: ¿Qué algoritmos permiten facilitar el desarrollo de un buscador?

Q2: ¿Qué procedimientos debe tener un un buscador?

Q3: ¿Cómo podemos evaluar el desempeño de los buscadores?

Para responder estas preguntas, se realizaron búsquedas de artículos en los siguientes bancos de datos: Science direct, Scopus, Springer, IEEE y ACM, usando las palabras clave necesarias como *domain specific, search, engine, document* y *retrieval*; además, se consideró el título, el resumen y las palabras clave.

Como criterio de exclusión, se limitaba el periodo de publicación desde 2010 hasta el 2018, y de los resultados se les daba prioridad a aquellos con factor de impacto SJR, pues SJR regula revistas estables y con impacto internacional. Además, se excluyeron aquellos artículos que se enfocaban en otro tipo de búsquedas (como la búsqueda web) y se evitaron aquellos artículos que repitieran un tema de un artículo ya seleccionado (se descartaban los más antiguos).

3.3 Ejecución

La implementación del protocolo establecido en la planificación se detalla a continuación en la figura 5, en donde se aprecia el proceso de selección de los artículos desde distintas fuentes de datos hasta la selección final.

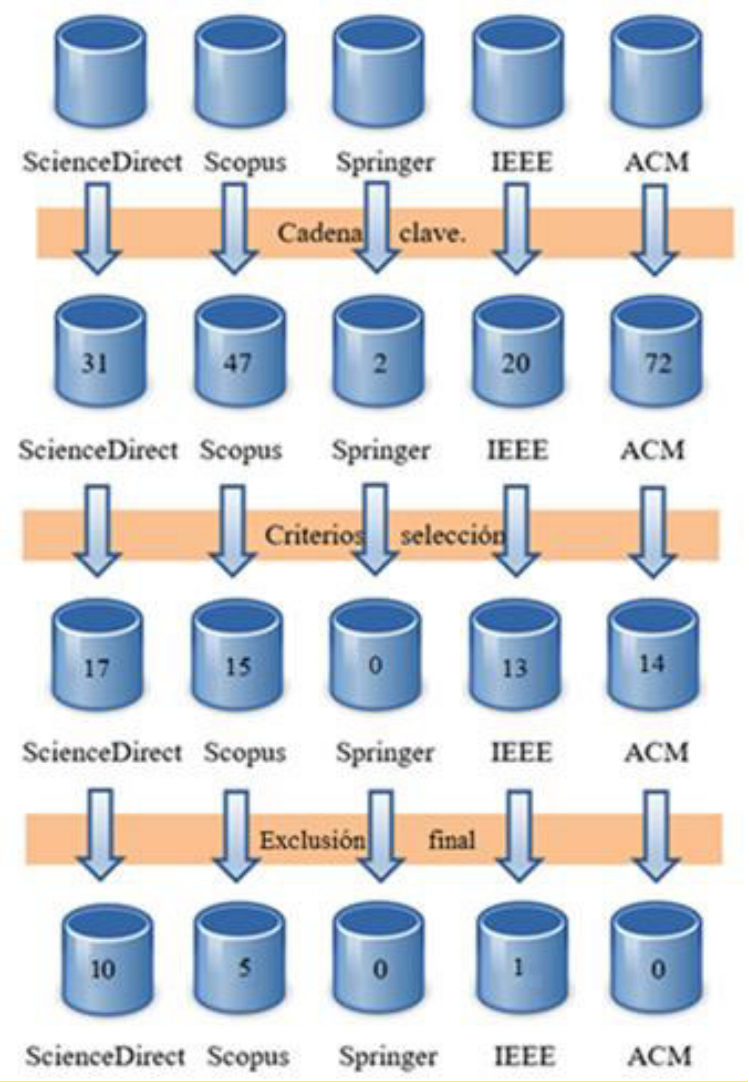


Figura 5. Diagrama de flujo de selección de artículos para el estado del arte

Se puede notar que, de los bancos de datos iniciales, se encontró inicialmente 172 artículos, y gracias a los criterios de exclusión establecidos se pudieron reducir a 59 artículos. Finalmente, de aquellos artículos que hablaban sobre un mismo tema, se seleccionaba solo uno (para no ser redundantes), siendo este el que tenga más detalle en el contenido, quedando 15 artículos.

3.4 Resultados

3.4.1 Artículos seleecionados

A continuación, en la tabla 1, se muestra la lista de actículos resultantes durante el proceso de selección.

Tabla 1. Lista de artículos usados en la investigación como parte del estado del arte

Id	Título del artículo	Autor(es)	Año	Revista
A01	CONQUIRO: A cluster-based meta-search engine	Maria Vargas-Vera, Tesica Castellanos y Miltiadis Lytras	2010	Computer in Human Behavior
A02	Development of Search Engines using Lucene: An Experience	Masnizah Mohd	2011	Procedia - Social and Behavioral Sciences
A03	Automatic classification of academic documents using text mining techniques	Haydemar Núñez y Esmeralda Ramos	2012	XXXVIII Conferencia Latinoamericana En Informatica
A04	A semantic similarity method based on information content exploiting multiple ontologies	David Sánchez y Monserrat Batet	2013	Expert Systems with Applications
A05	Searching Research Papers Using Clustering and Text Mining	Alan Calvillo, Alejandro Padilla y Jaime Muñoz	2013	International Conference on Electronics, Communications and Computing 2013
A06	A heuristic approach for k-representative information retrieval from large-scale data	Jin Zhang, Qiang Wei y Guoqing Chen	2014	Information Sciences
A07	Supporting information retrieval from electronic health records: A report of University of Michigan's nine-year experience in developing and using the Electronic Medical Record Search Engine (EMERSE)	David Hanauer, Qiaozhu Mei, James Law, Ritu Khanna y Kai Zheng	2015	Journal of Biomedical Informatics
A08	Textual Similarity based on Lexical-Semantic features	Alexander Chávez, Antonio Fernández, Héctor Dávila, Yoan Gutiérrez, Armando Collazo, José Abreu	2015	Second Joint Conference on Lexical and Computational Semantics

A09	An algorithm of finding thematically similar documents with creating context-semantic graph based on probabilistic-entropy approach	Moloshnikov I.A, Sboev A.G, Rybka R.B. y Gydovskikh D.V.	2015	Procedia Computer Science
A10	TMR: Towards an efficient semantic-based heterogeneous transportation media big data retrieval	Kehua Guo, Ruifang Zhang y Li Kuang	2016	Neurocomputing
A11	A semantic framework for textual data enrichment	Yoan Gutiérrez, Sonia Vázquez y Andrés Montoyo	2016	Expert Systems with Applications
A12	Text classification-based filters for a domain-specific search engine	Sebastian Schmidt, Steffen Schnitzer y Christoph Rensing	2016	Computers in Industry
A13	Efficient Indexing for Semantic Search	Fatemeh Lashkari, Faezeh Ensan, Ebrahim Bagheri, Ali A. Ghorbani	2016	Expert Systems with Applications
A14	Intelligent Predictive String Search Algorithm	Dipendra Gurung, Udit Kr. Chakraborty, Pratikshya Sharma	2016	Procedia Computer Science
A15	Assessing the impact of Stemming Accuracy on Information Retrieval –A multilingual perspective	Felipe N. Flores, Viviane P. Moreira	2017	Information Processing and Management
A16	Generation of simple structured information retrieval functions by genetic algorithm without stagnation	A.S. Kulunchakov, V.V. Strijov	2017	Expert Systems With Applications

3.4.2 Estadísticas

A continuación, en las figuras 6 y 7, se muestran gráficos que reflejan algunas relaciones entre los artículos seleccionados.

Proporción de artículos tentativos seleccionados y artículos tentativos excluidos

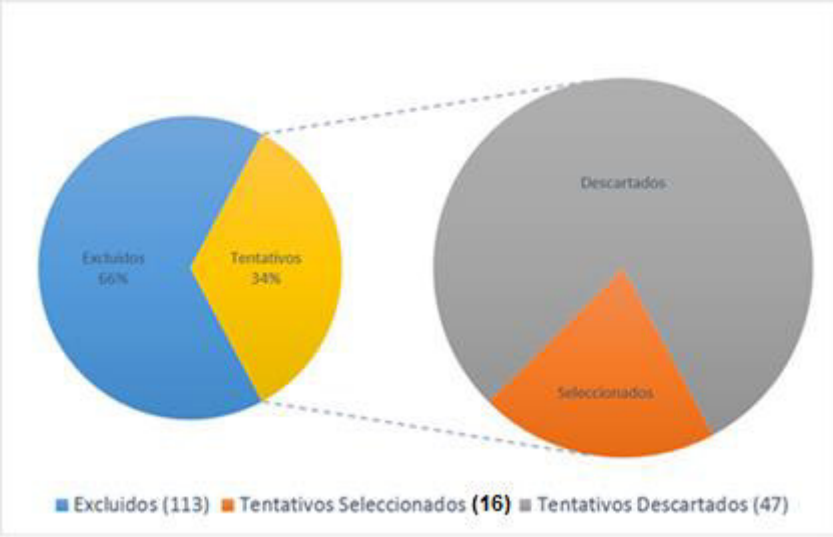


Figura 6. Relación entre artículos seleccionados y excluidos

Gráfica comparativa de factor de impacto SJR de las revistas de los artículos seleccionados

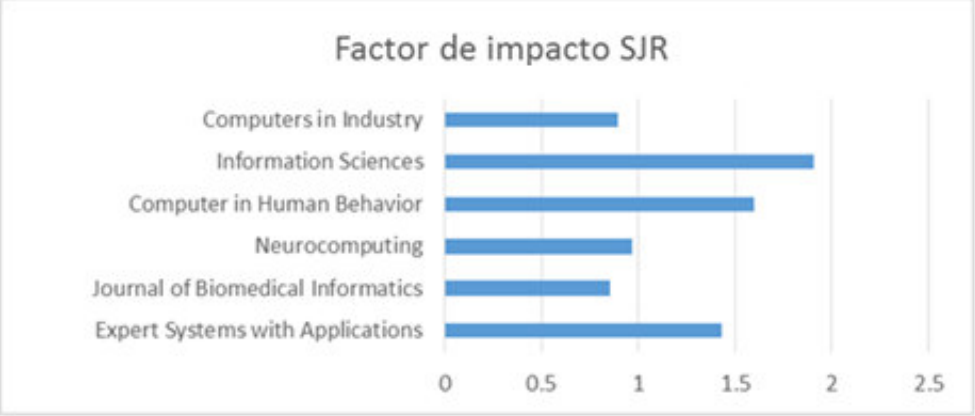


Figura 7. Factor de impacto SJR de las revistas seleccionadas

3.5 Análisis

En este apartado se responde a las preguntas planteadas durante la planificación, así como algunas conclusiones finales.

3.5.1 Algoritmos dentro de un buscador (Q1)

Se encontraron cinco (5) algoritmos para el realizar la búsqueda y recuperación de la información, de los cuales tres (3) son para el reconocimiento de patrones (Boyer Moore, Horspool y Knuth Morris Pratt) y dos (2) son para calcular la similitud entre dos patrones (Vector Space Model Score y Lucene's Practical Scoring).

Además, se resalta la indexación como solución a la necesidad de explorar un gran conjunto de documentos, para evitar la búsqueda en forma lineal en cada documento (Manning, Raghavan, & Schütze, 2008). En particular estos y otros autores mencionan que la indexación invertida se ha convertido en un estándar para la solución del problema de búsqueda y recuperación de documentos (Lashkaria, Ensanb, Bagheric, & Ghorbani, 2017) (Konow & Navarro, 2012) (Bast & Weber, 2006). El proceso de indexación será explicado en el subcapítulo 4.3.4.2, al explicar los componentes del sistema.

3.5.1.1 Búsqueda por coincidencia exacta

Gurung, Chakraborty y Sharma (2016) comparan los algoritmos de búsqueda de patrones por coincidencia exacta mencionados en el punto anterior, mencionando las ventajas de cada uno de ellos. A partir de dicha información, y de fuentes adicionales se ha elaborado la tabla 2

Tabla 2. Comparación de algoritmos de búsqueda de patrones por coincidencia exacta

Algoritmo	**Boyer Moore (BM)**	**Horspool**	**Knuth Morris Pratt (KMP)**
Comparación del Patrón	Empieza por el final de la cadena	Empieza por el final de la cadena	Empieza por el inicio de la cadena
Resumen	Si el último carácter no coincide, se desplaza según la tabla de prefijo malo. Si coincide, se desplaza según la tabla de sufijo bueno. La tabla del prefijo malo almacena la distancia de cada carácter al final del patron, mientras que la del sufijo bueno almacena la distancia entre subpatrones encontrados en el patrón de busqueda	Algoritmo BM simplificado, elimina la complejidad de BM al no considerar la tabla de sufijo bueno y solo la de prefijo malo	Reutilizar los subpatrones coincidentes dentro del patrón de búsqueda, previos al carácter de fallo para realizar los deplazamientos del patron dentro del texto, hasta que se encuentre la cadena o termine el texto
Comparador	Sufijo bueno y prefijo malo	Prefijo malo	Sufijo bueno
Ventaja	Eficiente para patrones grandes, es el más rápido algoritmo de búsqueda no indexado	Eficiente para alfabetos pequeños	Eficiente para grandes archivos, pero con patrones pequeños

3.5.1.2 Búsqueda por similitud

Mohd (2011) y Hanauer et al. (2015), desarrollan sus sistemas con los algoritmos de similitud Vector Space Model Score y Lucene's Practical Scoring, los cuales calculan un puntaje que viene a representar la similitud entre dos patrones (The Apache Software Foundation, 2016). En la tabla 3 se muestra un resumen de dichos algoritmos

Tabla 3. Comparación de algoritmos de búsqueda de patrones por coincidencia exacta

Algoritmo	**Vector Space Model Score**	**Lucene's Practical Scoring**
Descripción	Documentos y consultas son representados como vectores con peso, siendo cada termino una dimension, y el peso Tf-idf*	Boolean Model + VSM Score. Los documentos aprobados por el modelo boleano, son procesados por el VSM
Resumen	Similitud = Coseno de los vectores = $\frac{V(q) \cdot V(d)}{\lvert V(q)\rvert \, \lvert V(d)\rvert}$	Lucene refina VSM score: Se permite el uso de impulsos al documento, a las etiquetas, o a la consulta. Además, se controla cuando un documento coincide con muchos términos de la consulta sin coincidir todos, dando prioridad a aquellos documentos que tengan todas las coincidencias posibles
Ventaja	Menor complejidad	Mayor precisión

* Tf-idf	Frecuencia de un término sobre la cantidad de términos, multiplicado por el logaritmo de la razón de la cantidad de documentos del conjunto en los que aparece dicho termino

Pero ¿qué ventajas y desventajas existe en aplicar algoritmos de búsqueda de patrones por coincidencia exacta contra algoritmos de similitud? El reconocimiento de patrones exactos está optimizado para obtener solo coincidencias exactas, reduciendo el tiempo de respuesta en comparación a los algoritmos de similitud, pero sólo en estos casos particulares. Sin embargo, los algoritmos de similitud generan una salida que permite retroalimentar el resultado de una consulta, por ejemplo, se pueden establecer por coincidencias, a aquellas palabras que difieran de otras por una letra, cosa que de forma estadística sucede cuando el usuario comete errores ortográficos al momento de la búsqueda

3.5.2 Procedimientos dentro de un buscador (Q1)

Se encontraron cinco (5) procedimientos que, según los objetivos que se tenga para el buscador, se deben o no implementar en este. Estos son los siguientes: ordenamiento y presentación de los resultados, estructuración y almacenamiento del repositorio, eliminación de redundancia en los resultados, interpretación semántica, e indexación de los documentos. Además, se encontró una herramienta para el desarrollo de buscadores, que facilita la implementación de estos procedimientos: Solr (motor de búsqueda basado en la librería Lucene).

Para el ordenamiento y presentación de los resultados obtenidos, Vargas-Vera, Castellanos y Lytras (2010) desarrollaron CONQUIRO, una herramienta que como principal objetivo tenía organizar los documentos resultantes en grupos. Su principal contribución es haber ofrecido una manera eficiente con la que se puede dar sentido a los documentos recuperados en una consulta en

bloques. Luego, Moloshnikov, Sboev, Rybka, y Gydovskikh (2015) observaron que encontrar documentos de temática similar usando un grafo semántico probabilístico permite al sistema definir los temas relacionados al interés del usuario y sugerirlos durante la presentación de los resultados, concluyendo satisfactoriamente su utilidad, sobre todo para grandes colecciones de documentos.

En cuanto a la estructuración y almacenamiento del repositorio, Núñez y Ramos (2012) notaron la existencia de una gran cantidad de datos no estructurados en los medios, por lo que usando la minería de texto pudieron detectar patrones que permitieron predecir con una precisión del 88.9 % el área de conocimiento del documento en cuestión. Calvillo, Padilla y Muñoz (2013) detectaron que se pierde mucho tiempo buscando documentos cuando el buscador no puede mostrar los resultados que se quiere, por lo que ofrecen utilizar minería de textos para ayudar en la agrupación de artículos científicos, lo que resulta un aumento de la facilidad de búsqueda de artículos similares.

Por otro lado, en cuando a la eliminación de redundancia de los resultados, Zhang, Wei y Chen (2014) critican que la información que normalmente se obtiene en las búsquedas es muy redundante, y que la información relativa ligada a la consulta puede ser útil al usuario, por lo que al eliminar la redundancia obtienen que su sistema es más aceptado por el público por cubrir más información de utilidad. Kulunchakov y Strijov (2017) proponen un algoritmo genético que impone una penalidad a los documentos que tengan textos con superposición, es decir, que sean reiterativos. El resultado es un mejor conjunto de resultados, obteniendo dichos resultados significativamente más rápido que con otros métodos.

Luego, Felipe N. Flores, Viviane P. Moreira (2017) mencionan la lematización como parte de la interpretación semántica que se debe realizar a la consulta del usuario, pues los sufijos de género

y número, así como las palabras sin contenido semántico, deben ser evitadas, pues son usadas por el lenguaje natural, pero no aportan valor a la consulta. Por último, Lashkari, Ensan, Bagheri y Ghorbani (2016) hablan sobre la interpretación semántica y la indexación. Por el lado de la interpretación semántica, menciona los requisitos de almacenamiento y procesamiento que requiere tener la información semántica de cada uno de los conceptos del universo de documentos. También, explican la indexación invertida como la forma más eficiente de indexación de documentos a texto completo, pero no sin dejar de mencionar las desventajas de este tipo de estructura de datos.

Un resumen de los algoritmos existentes dentro de los buscadores es presentado en la tabla 4.

Tabla 4. Algoritmos dentro de un buscador

Procedimiento	**Descripción**	**Referencia**
Ordenamiento y presentación de los resultados	Tratamiento a los resultados obtenidos con la finalidad de priorizarlos de cada al usuario	Vargas-Vera, Castellanos y Lytras (2010) Moloshnikov, Sboev, Rybka, y Gydovskikh (2015)
Estructuración y almacenamiento del repositorio	Modo de almacenamiento del repositorio con la finalidad de permitir la búsqueda bajo ciertas condiciones particulares	Núñez y Ramos (2012) Calvillo, Padilla y Muñoz (2013)
Eliminación de redundancia en los resultados	Eliminar la redundancia de los resultados y/o sugerir alternativas	Zhang, Wei y Chen (2014) Kulunchakov y Strijov (2017)
Interpretación semántica	Tratamiento de la consulta ingresada por el usuario para obtener conceptos o ideas similares a lo consultado	Lashkari, Ensan, Bagheri y Ghorbani (2016) Flores, Moreira (2017)
Indexación de los documentos	Almacenamiento de la información relevante del universo de documentos para un fácil posterior acceso	Lashkari, Ensan, Bagheri y Ghorbani (2016)

Dos de los artículos encontrados (Mohd, 2011) (Hanauer, Mei, Law, Khanna, & Zheng, 2015) utilizan la herramienta Lucene para el desarrollo de buscadores. Destacan su capacidad de indexación por sobre otras herramientas, mediante el uso de la indexación invertida modificada de Lucena; alegaron que la indexación es el proceso más importante dentro de un buscador, pues las búsquedas se realizan sobre el dicho índice. Mohd (2011) también resalta a Lucene como una herramienta de fácil aprendizaje, tanto para estudiantes como para conocedores de alto nivel. Solr es mencionado por Hanauer, Mei, Law, Khanna, & Zheng (2015), pues su aplicación integraba Lucene y Solr para implementar la aplicación con un servidor de búsqueda y una aplicación web.

Las herramientas para la elaboración de los buscadores de los artículos profundizados en el estado del arte se muestran a continuación en la tabla 5.

Tabla 5. Herramientas para el desarrollo de buscadores y su aplicación

Herramienta	**Descripción**	**Aplicación**	**Referencia**	**Licencia**
Lucene	Herramienta Apache para el desarrollo de motores de búsqueda	Implementación de sistemas de búsqueda	Mohd (2011), Hanauer *et al.* (2015)	Libre
Solr	Implementación de motores de búsqueda web con Lucene	Elaboración del sistema EMERSE	Hanauer *et al.* (2015)	Libre

De la tabla 5 se concluye que la herramienta por excelencia para el desarrollo de buscadores a la fecha es Lucene, pues en su mayoría los autores que desarrollaban un sistema de búsqueda indicaban que usaron esta herramienta. En la tabla 6 se detalla los procedimientos que implementan Solr y Lucene, referenciando los artículos que los mencionan.

Tabla 6. Herramientas y los procedimientos mencionados en los artículos

Herramienta	Procedimientos	Referencia
Lucene y Solr	- Indexación - Ordenamiento de resultados - Búsqueda de múltiples índices - Agrupamiento de resultados - Búsqueda por etiquetas - Tokenización - Lematización - Búsqueda filtrada - Diccionarios semánticos	Mohd (2011), Hanauer *et al.* (2015)

3.5.3 Evaluación de desempeño (Q3)

Usualmente, las métricas para la evaluación de desempeño de los sistemas de búsqueda son la precisión y la exhaustividad. La precisión consiste en la fracción de documentos recuperados que son relevantes para la cadena de búsqueda; mientras que la exhaustividad es la fracción de documentos relevantes que han sido recuperados (Martinez & Rodriguez, 2004) (Manning, Raghavan, & Schütze, 2008).

$$Precisión = \frac{|\{documentos\ relevantes\} \cap \{documentos\ recuperados\}|}{|\{documentos\ recuperados\}|}$$

$$Exhaustividad = \frac{|\{documentos\ relevantes\} \cap \{documentos\ recuperados\}|}{|\{documentos\ relevantes\}|}$$

Además, existen otras maneras de evaluar un sistema, por ejemplo, la percepción o satisfacción de los usuarios finales (Liu & Guo, 2008), los cuales se pueden dar por la satisfacción del usuario con el sistema (Armstrong, Fogarty, Dingsdag, & Dimbleby, 2005), intención de uso por parte del

usuario (Rana, Dwivedi, Williams, & Weerakkody, 2015), por el tiempo de respuesta o por las funcionalidades que al usuario le parezcan útiles en un momento dado. Estas características de satisfacción solo pueden ser medidas mediante una encuesta de escala numérica, pues la satisfacción no es una variable cuantitativa.

Otros autores solo se enfocan en mejorar algún valor en comparación con los resultados obtenidos por otro sistema, algoritmo o técnica. Así, en la tabla 7 se puede apreciar el resultado que obtuvo o mejoró cada buscador y/o algoritmo repasado en el estado del arte.

Tabla 7. Evaluación de desempeño por cada artículo

Algoritmo o Procedimiento	**Desempeño**	**Referencia**
Organización de documentos	Hasta 86 % de precisión y 41 % de exhaustividad	Maria Vargas-Vera, Tesica Castellanos y Miltiadis Lytras (2010)
Minería de Texto	95,65 % de documentos clasificados correctamente	Haydemar Núñez y Esmeralda Ramos (2012)
Eliminación de Redundancia	82 % de precisión, superando a las herramientas comparadas	Kehua Guo, Ruifang Zhang y Li Kuang (2014)
Lematización	Hasta 73 % de precisión en palabras en Español y 79 % en inglés	Felipe N. Flores, Viviane P. Moreira (2017)

CAPÍTULO 4: DISEÑO Y DESARROLLO DEL BUSCADOR ESPECIALIZADO

En este capítulo se presenta un sistema de búsqueda desarrollado a través de la metodología Scrum. Se analiza la problemática y los requerimientos necesarios para implementar una propuesta que aporte valor al usuario final, así, se da solución a las necesidades de búsqueda de legislación peruana sobre tecnología de información.

4.1 Metodología de desarrollo

Los proyectos de desarrollo de software necesitan una metodología que permita estructurar, planificar y controlar todo el proceso de desarrollo, y nuestro buscador no es la excepción. Ahora, ¿qué opciones existen y cuál de ellas conviene utilizar? Se tiene las metodologías tradicionales y las metodologías ágiles. Las metodologías tradicionales buscan predecir las posibles tareas que puedan surgir durante todo el proyecto, redactando detalladamente cada fase del proyecto y sus procesos, resultando una documentación de mayor calidad, pero, como consecuencia, el tiempo requerido para el desarrollo es mucho mayor, haciendo que en ocasiones se reduzca el tiempo asignado a otros procesos como a las pruebas. Por otro lado, en las metodologías ágiles se da por supuesto que existirán variables no controladas o que pueden cambiar, así, se necesita un control permanente en lo que se ha hecho y lo que falta por hacer, de manera iterativa o incremental, teniendo en estos casos una documentación más escasa porque lo primordial es obtener un producto funcional.

Un buscador de legislación de tecnología de información es un proyecto poco usual, pues, como se ha mostrado, no existe ningún otro buscador de este tipo en el mercado. Es por ello que se desconoce las necesidades de los usuarios, es decir, el alcance es propenso a sufrir modificaciones,

siendo por este principal y determinante motivo que se optó por seleccionar alguna metodología de desarrollo ágil. Ahora, ¿cuál metodología ágil es la más adecuada para este escenario? Existen muchas investigaciones que hacen comparaciones entre las diversas metodologías, estableciendo sus marcos de trabajo (Chan & Thong, 2009), o analizando la satisfacción de los usuarios de estas metodologías (Lei, Ganjeizadeh, Kumar, & Ozcan, 2015). Todas ellas reconocían que las metodologías ágiles más usadas y recomendadas son Extreme Programming (XP) y Scrum. Se diferencian en que XP se basa en el desarrollo, mientras que Scrum puede incluir documentación como productos, por lo que también considera la gestión del proyecto. Además, la metodología Scrum ya ha sido aplicada en una gran variedad de proyectos debido a que se puede monitorear el avance del desarrollo con mayor facilidad (Scrum Alliance, 2016), siendo otros factores claves para la elección de esta metodología sobre otras, la presencia de un equipo de desarrollo pequeño y el tiempo de desarrollo limitado (Korteweg, 2016). Por todo ello, para el desarrollo de nuestro sistema, se optó por la metodología ágil de desarrollo Scrum.

En resumen, para el desarrollo del buscador, se plantea usar Scrum, ya que permite adaptarse y responder más rápidamente y con mayor precisión a los cambios que inevitablemente aparecen durante el desarrollo de un proyecto de software (Scrum Alliance, 2016).

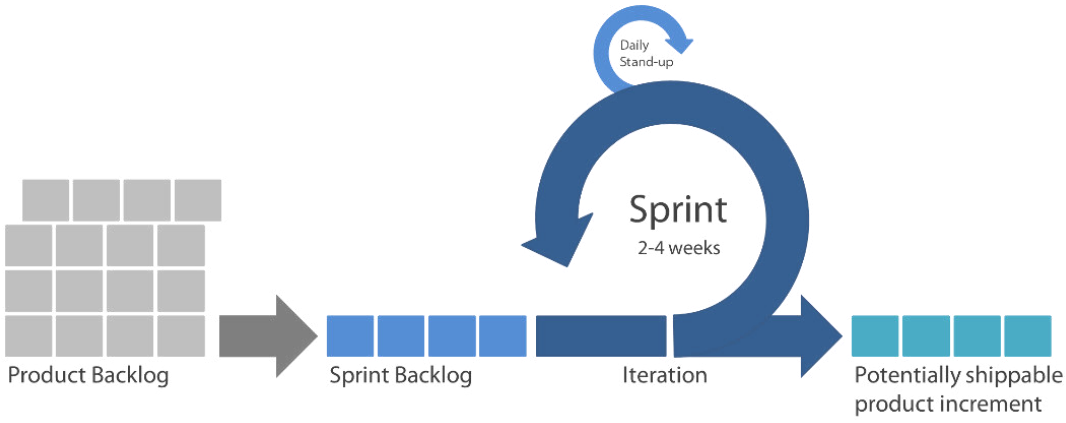


Figura 8. Diagrama general de Scrum

Fuente: Documentación de Scrum

La figura 8 contiene el diagrama general de Scrum. Al lado izquierdo tenemos las historias de usuario agrupadas en el Product Backlog. De ellas se seleccionan algunas para cada uno de los Sprints, que a su vez tienen una retroalimentación diaria. Al completarse la iteración, es posible modificar alcances, condiciones u otros en las siguientes iteraciones.

4.2 Desarrollo

El primer paso es establecer el equipo de trabajo, siendo el Scrum Manager el Doctor en Ciencias de la Computación David Mauricio; también se establece quién será el cliente, en este caso será el Doctor en Derecho Informático Carlos Ferreyros. Ya establecidos los roles de desarrollo del proyecto, se procede a realizar la creación de un Product Backlog, que contiene la lista de requisitos del proyecto desde la perspectiva del cliente (historias de usuario). Estos requisitos buscan solucionar la problemática existente en otros sitemas de búsqueda, además de establecer características que se desea que el sistema de busca tenga. Estos temas se establecen en un acta de constitución del proyecto durante una reunión inicial que se explicará a continuación. Se adjunta la evidencia de esta reunión de constitución del proyecto en el Anexo A-1.

4.2.1 Reunión inicial

La reunión inicial del proyecto se realiza con el objetivo de analizar los requerimientos del sistema a desarrollar (análisis de requerimientos), a modo de historias de usuario. Esto permite la creación de un primer modelo de Product Backlog que posteriormente, según las necesidades y problemáticas que se vayan presentando durante el desarrollo, se puede actualizar (añadir, eliminar, modificar) en cuanto a los requerimientos que inicialmente fueron planteados en él.

4.2.1.1 Análisis de requerimientos

Para obtener las necesidades de los potenciales usuarios del sistema, se investigó en diferentes estudios jurídicos sobre las limitaciones que tenían los sistemas que actualmente usaban, entre los cuales se identificó el Sistema Peruano de Información Jurídica (SPIJ, 2017) y el Archivo Digital de la Nación del Congreso de la República (ADNCR, 2017)

Por un lado, el buscador del Archivo Digital de la Nación del Congreso de la República permite buscar por rango de fecha y tipo de legislación a través del uso de palabras clave; y por el otro lado, el buscador denominado Sistema Peruano de Información Jurídica (SPIJ) (http://spij.minjus.gob.pe/) contiene información acerca de la Constitución política, normas, leyes orgánicas, entre otros, logrando cubrir gran parte de la legislación vigente.

Durante la reunión de constitución del proyecto, formalizado en el acta del mismo nombre (ver Anexo A-1), se detectaron siete (7) problemas o limitaciones en estos sistemas. Dichos problemas son presentados en la tabla 8.

Tabla 8. Problemática detectada

Id	Problemática
P01	Los resultados obtenidos son diferentes si utilizas la misma palabra con o sin tilde.
P02	Si se comete un error ortográfico, el buscador no encontrará lo que se quiso buscar.
P03	No se puede buscar en varias categorías de leyes a la vez.
P04	No se puede descargar la legislación como pdf (en SPIJ), pues la legislación se encuentra incrustada en la web (en HTML).
P05	Falta vista rápida o resumen para comprobar que el documento efectivamente tiene lo buscado en el contexto que se necesita.
P06	La legislación está organizada por fechas y por categorías, pero es posible combinar categorías o rangos de fecha.
P07	La interfaz gráfica de usuarios es complicada, ya que de entrada presenta demasiadas opciones que un usuario promedio no usa.

Las características que tendrá nuestro sistema deben solucionar la problemática detectada; para la facilidad del trabajo, se agrupó las características en tres áreas: características de precisión y exhaustividad, características de dominio específico, y características de experiencia de usuario, tal como se muestran en la tabla 9. Esta agrupación de la problemática existente permitirá dar una idea de cómo abordar el desarrollo de la propuesta que se va a implementar.

Tabla 9. Características establecidas para solucionar la problemática

Id	Característica	Problemas a solucionar
C1	Precisión y exhaustividad	P01, P04
C2	Dominio específico	P02, P05, P06
C3	Experiencia de usuario	P03, P07

4.2.1.2 Acuerdos iniciales

Después de levantados los requerimientos, se debía definir cómo afrontar el desarrollo del buscador, considerando que el desarrollo debe realizarse de tal manera que permita probar el desempeño del proceso de búsqueda propiamente dicho y cumpla con los requerimientos solicitados.

En primer lugar, este proyecto a priori no requiere un lenguaje de programación en específico, por ende, para seleccionar las tecnologías a utilizar, se partió de la premisa de conocer cuál era el lenguaje de programación que el equipo de trabajo conocía mejor: Java; en particular, java 8.0. Establecido el lenguaje de programación principal, se investigó en el mercado cuáles eran las tecnologías actuales usadas para el desarrollo, tanto para el buscador como para la aplicación web, tomando en cuenta el análisis que se hizo durante la revisión de la literatura de las herramientas para el desarrollo de buscadores más recomendadas (Q2, subcapítulo 3.5.2); se debe recordar que en ambos casos debían ser tecnologías compatibles con Java para acelerar el desarrollo. Así, para el buscador se seleccionó Solr (en concreto Solr 7.5), motor de búsqueda basado en Lucene, que

es una librería con licencia Apache 2.0, escrita en Java (The Apache Software Foundation, 2016), mientras que para el diseño y construcción de la aplicación web se seleccionó Vaadin (en concreto Vaadin 10.0), que es un Framework para el desarrollo de aplicaciones web Java, el cual permite utilizar dicho lenguaje de programación para crear interfaces de usuario en HTML5.

Apache Solr fue una de las herramientas detalladas en el punto 3.5.2, siendo usada por EMERSE (Hanauer, Mei, Law, Khanna, & Zheng, 2015), así que tenía probada validez en desarrollos de este tipo. Así, no solo se logró alinear las tecnologías usadas por otros investigadores, sino que también se logró la integración de Java en todos los componentes del desarrollo, permitiendo que el equipo de trabajo pueda dedicarse a tareas multidisciplinarias.

4.2.1.3 Product Backlog

Se define Product Backlog a partir de los requerimientos del cliente, identificando su aporte al sistema y, por tanto, sus prioridades; así como el costo estimado de desarrollo de cada una de ellas. Para ello se han usado valores desde 0 hasta 1000, escala en la que 0 significa sin costo (o valor) alguno, y 1000, un costo (o valor) máximo.

Tabla 10. Product Backlog final de DoLaw

Id	Descripción	Coste estimado	Condiciones de satisfacción	Valor aportado
HU1	Como cliente, deseo poder visualizar de manera gráfica cómo funciona el sistema	600	Diferenciar las características del administrador con las del usuario, en un diagrama de arquitectura	700
HU2	Como cliente, deseo que la legislación sea almacenada en pdf	800	Debe mantenerse los archivos en formato pdf	600
HU3	Como cliente, deseo que el producto tenga un control de acceso (Login) en caso deba ingresar como administrador	400	El login debe consistir de id y contraseña y dará accesos de administrador. Un usuario regular no necesitará login	300
HU4	Como administrador, deseo registrar nueva legislación al sistema	500	Los documentos de legislación deben poder ingresarse mediante el sistema	800

HU5	Como administrador, deseo actualizar la legislación ingresada.	600	Los documentos de legislación deben poder actualizarse cuando se desee	700
HU6	Como administrador, deseo un diseño de interfaz gráfica agradable, para realizar mantenimiento	300	Debe aprobarse el maquetado realizado antes de realizar la implementación del diseño	800
HU7	Como usuario, deseo un diseño de interfaz de búsqueda fácil de usar	400	Debe aprobarse el diseño realizado antes de realizar la implementación.	900
HU8	Como usuario, deseo que el sistema busque la documentación que coincida con las palabras clave ingresadas, y con sus sinónimos	1000	Debe permitirse palabras o frases clave para realizar la búsqueda, y que la búsqueda se haga por sus sinónimos	1000
HU9	Como usuario, deseo recibir con precisión los resultados obtenidos	900	Debe haber un tratamiento especial de las consultas para mostrar la mayor cantidad de resultados relevantes sobre el total de resultados	900
HU10	Como usuario, deseo filtrar la legislación por categorías o fecha de publicación	900	Se debe ofrecer el filtrado por categoría y fecha	800
HU11	Como usuario, deseo poder descargar la legislación correspondiente a los resultados obtenidos o elegidos	200	La descarga de los resultados debe hacerse en pdf, además de la vista previa	500
HU12	Como usuario, deseo que los resultados de mi búsqueda estén ordenados por prioridad	300	La prioridad será definida colocando primero el documento con mayor relevancia con la consulta	300
HU13	Como usuario, deseo que el sistema contenga siglas y abreviaciones correspondientes al área legislativa	800	La búsqueda debe poder realizarse tanto buscando las siglas o abreviación, como buscando la palabra completa	800
HU14	Como usuario, deseo que al ingresar una consulta, se ignore sufijos de género y número en las palabras ingresadas	800	El sistema debe obtener las raíces de las palabras ingresadas y realizar la búsqueda mediante estas	900
HU15	Como usuario, deseo que los resultados se obtengan en un tiempo de respuesta menos a 2 segundos	600	El tiempo de respuesta del servidor será de dos segundos. Además, se plantea una interfaz que reduzca el tiempo de carga de la web desde internet	500
HU16	Como usuario, deseo que el sistema entienda lo que estoy buscando (búsqueda semántica), incluso si cometo un error ortográfico	900	El sistema debe buscar palabras similares a las ingresadas; y debe hacer un tratamiento de la consulta para realizar subconsultas que el usuario pueda necesitar a partir de lo ingresado	1000

La tabla 10 muestra el Product Backlog final con las historias de usuario de nuestro buscador, corregido después de las iteraciones de los sprints del desarrollo (la versión inicial y los sprints se puede encontrar en el Anexo A, así como la fecha límite asignada para las tareas e iteraciones).

4.2.2 Sprint 1

Se decidió que, en el primer sprint (Anexo A-2), se realicen tareas de modelamiento y arquitectura del sistema, es decir, se busca representar la problemática en una arquitectura y modelo de base de datos que soporten la solución. La tabla 11 muestra las historias seleccionadas para esta iteración.

Tabla 11. Sprint 1

Id	Descripción
HU1	Como cliente, deseo poder visualizar de manera gráfica cómo funciona el sistema
HU2	Como cliente, deseo que la legislación sea almacenada en pdf
HU3	Como cliente, deseo que el producto tenga un control de acceso (Login) en caso deba ingresar como administrador

4.2.2.1 Desarrollo de las historias de usuario

HU1: Se realizó un diagrama de arquitectura y un diagrama de flujo para explicar el funcionamiento del sistema. Estos diagramas se desarrollaron con LucidChart, una herramienta online de creación de diagramas, el cual incluye BPMN 2.0, nomenclatura que fue solicitada durante la reunión del sprint. Dichos diagramas están incluidos en el subcapítulo 4.3, en el que se habla del buscador, en los puntos 4.3.1 y 4.3.2, respectivamente (ver figuras 9 y 10).

HU2: Durante la fase de pruebas y validación, se almacenará en un servidor de archivos, trescientos diecisiete (317) archivos de legislación peruana de TI, del período entre el 2000 y el 2018, base que fue proporcionada por el estudio jurídico Ferreyros & Ferreyros, todas en formato pdf. Se destaca que la problemática P04 se soluciona en esta historia de usuario, al permitir la carga de documentos en formato pdf

HU3: El control de acceso se desarrolla mediante acceso por Gmail, utilizando la API de Google, que permite la obtención de los datos personales bajo consentimiento del usuario, o por ingreso de correo y contraseña.

4.2.2.2 Observaciones sobre el sprint 1

Las observaciones del Sprint 1 se realizan como primer tema a tocar en el Acta del Sprint 2 (ver Anexo A-3). En este caso, el primer sprint no tuvo observaciones.

4.2.3 Sprint 2

El segundo Sprint (Anexo A-3) incluye tareas para el administrador del sistema, las cuales se pueden observar en la tabla 12.

Tabla 12. Sprint 2

Id	Descripción
HU4	Como administrador, deseo registrar nueva legislación al sistema
HU5	Como administrador, deseo actualizar la legislación ingresada
HU6	Como administrador, deseo un diseño de interfaz gráfica agradable, para realizar mantenimiento

4.2.3.1 Desarrollo de las historias de usuario

HU4: El registro de nueva legislación debe realizarse a través del propio sistema, el cual debe permitir que se pueda definir la fecha de publicación y el tipo de ley de cada uno de las leyes que se carguen, incluso si esta carga se hace en bloque; de hecho, para hacer las pruebas se hizo la carga inicial de las 317 leyes a través de la interfaz de administrador realizada.

HU5: Al igual que existe una carga masiva, debe existir una interfaz de actualización. Por motivos de usabilidad, el diseño de ambas debe ser similares, así como las características de, en este caso, actualización en bloque. De hecho, en ambos se establece el primer paso para la solución a la problemática P03 y P06, pues se está realizando una clasificación de la legislación al momento de la carga, para luego hacer la consulta respectiva a la legislación clasificada.

HU6: Como se mencionó en la HU4, la carga de legislación nueva debe hacerse a través del sistema, y para ello la interfaz debe ser amigable. La interfaz de carga se presentará en el subcapítulo 4.3.5, en donde se muestran todas las interfaces gráficas del buscador.

4.2.3.2 Observaciones sobre el sprint 2

Se dieron sugerencias en el maquetado inicial del proyecto. La historia HU6 se corregirá en esta iteración, considerando que se debe aprobar un maquetado de la interfaz antes de realizar el desarrollo.

4.2.4 Sprint 3

El tercer sprint (Anexo A-4) contiene tareas de desarrollo orientadas a un buen diseño de interfaz gráfica para el usuario del sistema, así como las funcionalidades de estas interfaces. Las historias correspondietes se presentan en la tabla 13.

Tabla 13. Sprint 3

Id	Condiciones de satisfacción
HU6 *	Debe aprobarse el maquetado realizado antes de realizar la implementación del diseño
HU7	Debe aprobarse el diseño realizado antes de realizar la implementación
HU8	Debe permitirse palabras o frases clave para realizar la búsqueda, y que la búsqueda se haga por sus sinónimos

4.2.4.1 Desarrollo de las historias de usuario

HU6: Siguiendo las correcciones hechas al cierre del segundo Sprint, el maquetado principal del sistema se presenta en el punto 4.3.4.7 (figura 24), en donde se muestra la forma en la que se propone presentar los resultados. Luego, la interfaz gráfica final que nace a partir de este maquetado se muestra en el punto 4.3.5 (figuras 25, 26, 27 y 28).

HU7: La interfaz debe proporcionar de manera limpia toda la información que el usuario requiere; para ser precisos, aquí se solventará la problemática P05 y P07. Los diseños de las interfaces son presentados en el punto 4.3.5, mientras que como se dijo en los acuerdos iniciales, se utilizó Vaadin para el diseño de estas.

HU8: El algoritmo de búsqueda tendrá un componente que interpretará la cadena de búsqueda ingresada por el usuario, hallará en primer lugar los sinónimos de las palabras, y luego solicitará al índice las coincidencias de las palabras y sus sinónimos. Solr es capaz de ser configurado para acceder a diccionarios de equivalencias en formato txt, que puede ser usado para siglas, sinónimos, antónimos, o en lo que se requiera. ¿Por qué debemos agregar antónimos? Pues la relación de antonimia, si se aplica en casos como “no satisfecho” e “insatisfecho”, significa lo mismo, por lo

que sería útil buscar ambas palabras. Este proceso se explica en el punto 4.3.4.4, en la generación de consultas.

4.2.4.2 Observaciones sobre el sprint 3

En este caso, se dieron por levantadas las observaiones que se hicieron en el anterior sprint sobre la historia HU6, y no hubo nuevas observaciones.

4.2.5 Sprint 4

Para el cuarto sprint (Anexo A-5), se escogieron características relevantes para el sistema, es por ello que a este sprint se le asignó tres meses, un tiempo mayor al promedio cada sprint, de un mes. Debe recordarse que el detalle de las fechas de entrega establecidas quedó acordada y plasmada en el acta de sprint correspondiente. Las historias de este sprint son las mostradas en la tabla 14.

Tabla 14. Sprint 4

Id	Descripción
HU9	Como usuario, deseo recibir con precisión los resultados obtenidos
HU10	Como usuario, deseo filtrar la legislación por categorías o fecha de publicación
HU11	Como usuario, deseo poder descargar la legislación correspondiente a los resultados obtenidos o elegidos
HU12	Como usuario, deseo que los resultados de mi búsqueda estén ordenados por prioridad

4.2.5.1 Desarrollo de las historias de usuario

HU9: Para mostrar precisión, se debe obtener una gran cantidad de resultados relevantes con respecto a los resultados obtenidos. En este caso, para el sistema propuesto, hay que tener precaución en no hacer demasiados reemplazos por sinónimos o siglas, pues este proceso puede afectar la precisión, al aumentar los resultados obtenidos.

HU10: Para poder filtrar la legislación, inicialmente se hizo la carga considerando las categorías solicitadas. Luego, el diseño de la interfaz debe permitir filtrar tanto antes como después de realizada la búsqueda; en el primer caso el filtro se encuentra sobre el conjunto de documentos, y en el segundo caso, en el conjunto de resultados. En esta historia se terminan de solventar las problemáticas P03 y P06, cuyo proceso fue iniciado en la clasificación hecha por el administrador (Historia HU4 y HU5). En este caso, Solr te permite establecer etiquetas por cada documento, que son estucturas de datos donde se almacena una descripción del documento; es así que definimos una etiqueta Fecha de publicación, una de Jerarquía, y una Temática, además de las etiquetas que existen por defecto, entre ellas, la etiqueta que genera un id, la que almacena el título, y otra con el propio contenido de los documentos.

HU11: Durante el diseño debemos contemplar una opción que permita al usuario descargar la legislación correspondiente a los resultados que le aparecen en pantalla; y como las leyes fueron cargadas en PDF, le permitiremos al usuario descargarla en este mismo formato. Aquí estamos dando solución a la problemática P04.

HU12: Mostrar los resultados obtenidos de una consulta de forma ordenada dará una mayor percepción al usuario de que los documentos que se están obteniendo tienen alta precisión. De hecho, Solr, al estar basado en Lucene, posee su sistema de puntuación, el cual a través de criterios

de coincidencia ponderada establece un valor numérico entre la consulta y cada uno de los documentos, haciendo así posible su ordenamiento.

4.2.5.2 Observaciones sobre el sprint 4

No hubo observaciones sobre este sprint, porque deseaban ver las funcionalidades desarrolladas en este sprint integradas con las que venían a continuación.

Sin embargo, se solicitó una tarea adicional. Para lograr un aporte mayor y más diferencial, se pidió incluir la búsqueda semántica, que consiste en buscar no solo lo que escribe el usuario, sino lo que quiso decir. Dicha característica se la agregó en la HU16, la cual inicialmente no estaba planteada en el product backlog.

4.2.6 Sprint 5

Este es el último sprint (Anexo A-6), y al igual que el anterior, incluye el desarrollo de componentes importantes para el buscador, y también se dio el mismo plazo que el sprint 4: tres meses. A continuación mostramos las historias de este sprint en la tabla 15.

Tabla 15. Sprint 5

Id	Descripción
HU13	Como usuario, deseo que el sistema contenga siglas y abreviaciones correspondientes al área legislativa
HU14	Como usuario, deseo que al ingresar una consulta, se ignore sufijos de género y número en las palabras ingresadas
HU15	Como usuario, deseo que los resultados se obtengan en un tiempo de respuesta menos a 2 segundos
HU16 *	Como usuario, deseo que el sistema entienda lo que estoy buscando (búsqueda semántica), incluso si cometo un error ortográfico

4.2.6.1 Desarrollo de las historias de usuario

HU13: Para poder realizar la transformación de siglas por su significado textual, es necesario definir un diccionario que pueda ser leído por Solr, al igual que se hizo en el mapeado de sinónimos en la HU8, solo que en este caso una sigla tendrá por "sinónimo" una frase.

HU14: Los sufijos de una palabra son los que definen género y número. Para evitar que la búsqueda traiga diferentes resultados cuando el usuario busca la misma palabra en singular y en plural, se debe procesar la consulta con el fin de extraer la raíz de las palabras consultadas, y buscar la raíz, mas no la palabra completa. Para que este proceso sea fructífero, durante la indexación también se debe realizar la extracción de las raíces de las palabras, ya que de otro modo no se encontrarían las coincidencias deseadas.

HU15: El tiempo de respuesta es afectado por dos principales motivos: la indexación y el procesamiento de la consulta. Solr posee una buena indexación, en particular usa la indexación invertida, la cual almacena el mapeo de cada palabra del contenido de un texto con sus ubicaciones en el conjunto de textos. Muchos otros estudios usan esta técnica de Solr, como Hanauer (2015). Por otro lado, el procesamiento de la consulta no debe ser excesivo, pues afectará el tiempo de respuesta, por lo que se monitoreó las modificaciones que se realizaban y su impacto en el tiempo de respuesta, es por ello que se buscará como máximo dos equivalencias (sea sinónimos, antónimos o siglas) por palabra consultada.

HU16: Se notó que muchas características implementadas eran parte de una interpretación semántica no muy profunda, así que para terminar de atender las problemáticas pendientes (P01, P02) se culminó el procesamiento semántico de consultas. Para más detalle tenemos el componente Generar Query, en el punto 4.3.4.4.

4.2.6.2 Observaciones sobre el sprint 5

Al ser el último sprint, las observaciones se dieron hasta tener la satisfacción total del producto, para luego proceder a realizar el acta de cierre de proyecto, dando por aceptadas las características del mismo. El Acta de cierre también está adjunta en los anexos (Anexo A-7).

4.2.7 Cierre del Proyecto

El cierre del proyecto dio conformidad a las características, tanto aportes como limitaciones, del proyecto realizado. Aquí, al señalar las limitaciones, también se establecieron los trabajos futuros que en el capítulo correspondiente serán detallados (Capítulo 6). El acta de cierre se adjunta en el Anexo A-7. Con el proyecto terminado, se decidió bautizarlo como DoLaw.

4.3 Buscador

A continuación se dará a conocer con mayor detalle los principales entregables y componentes del buscador especializado de legislación peruana de tecnología de información DoLaw, así, se explicará el contenido de forma más técnica. El sistema debe solucionar los problemas detectados a través de las tres características planteadas, por lo que a continuación se dará una propuesta que soporte la solución con las características deseadas. El desarrollo se hizo en una computadora Intel Core i7 4720HQ a 2.60Ghz, con 16GB de Ram, de 64 bits, con Java 8.0 instalado. Estos requerimientos son los que tendría que tener el servidor de búsqueda para obtener resultados de respuesta similares, pero el usuario podrá acceder con cualquier navegador que soporte Java 8.

4.3.1 Arquitectura del sistema

La arquitectura del sistema muestra los componentes principales y sus conexiones entre sí. A continuación, en el diagrama de la arquitectura de DoLaw de la figura 9, se puede ver los servidores definidos y su contenido, así como la forma de conexión con el servidor, que en este caso es a través de una aplicación web, usando un navegador y permitiendo, así, el acceso desde cualquier dispositivo.

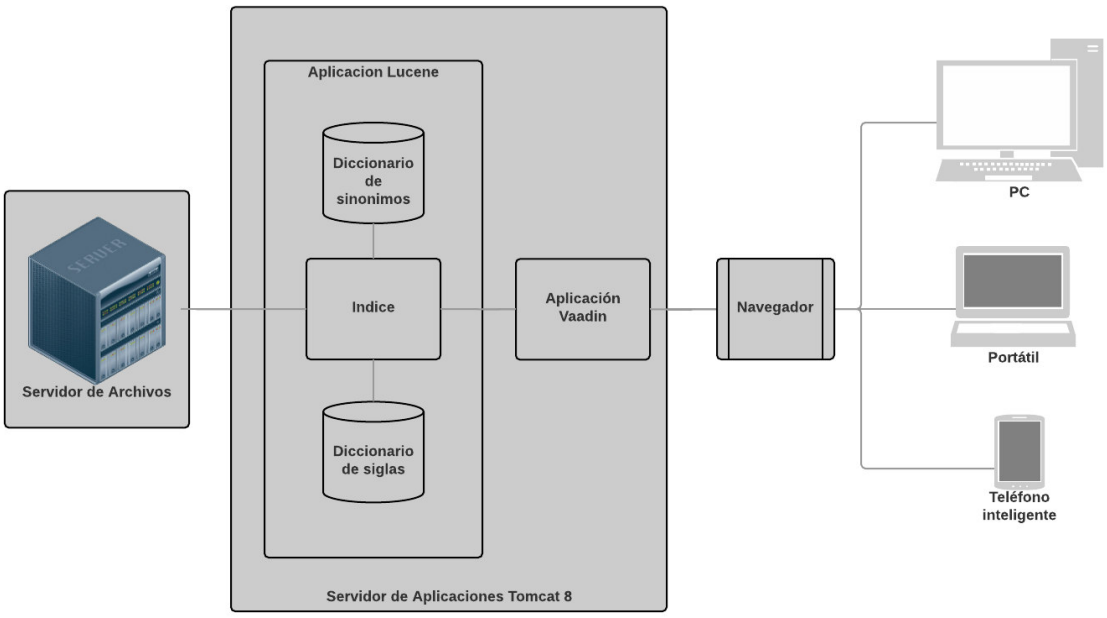


Figura 9. Diagrama de Arquitectura

Para el desarrollo se escogió utilizar Web Server for Chrome 0.4.8, una aplicación de Chrome que permite simular una carpeta de la computadora o servidor como servidor de archivos, utilizando HTTP. Para el despliegue, se utilizará una máquina virtual Windows con 1TB de almacenamiento y 2GB de RAM.

4.3.2 Diagrama de Flujo

A continuación, en la figura 10, se presenta el diagrama de flujo de DoLaw, en el cual se puede ver a los dos roles del sistema, usuario y administrador, y sus respectivos flujos de proceso, así como también el procedimiento que sigue el propio sistema cada vez que es usado por algún actor, sea de rol usuario o administrador.

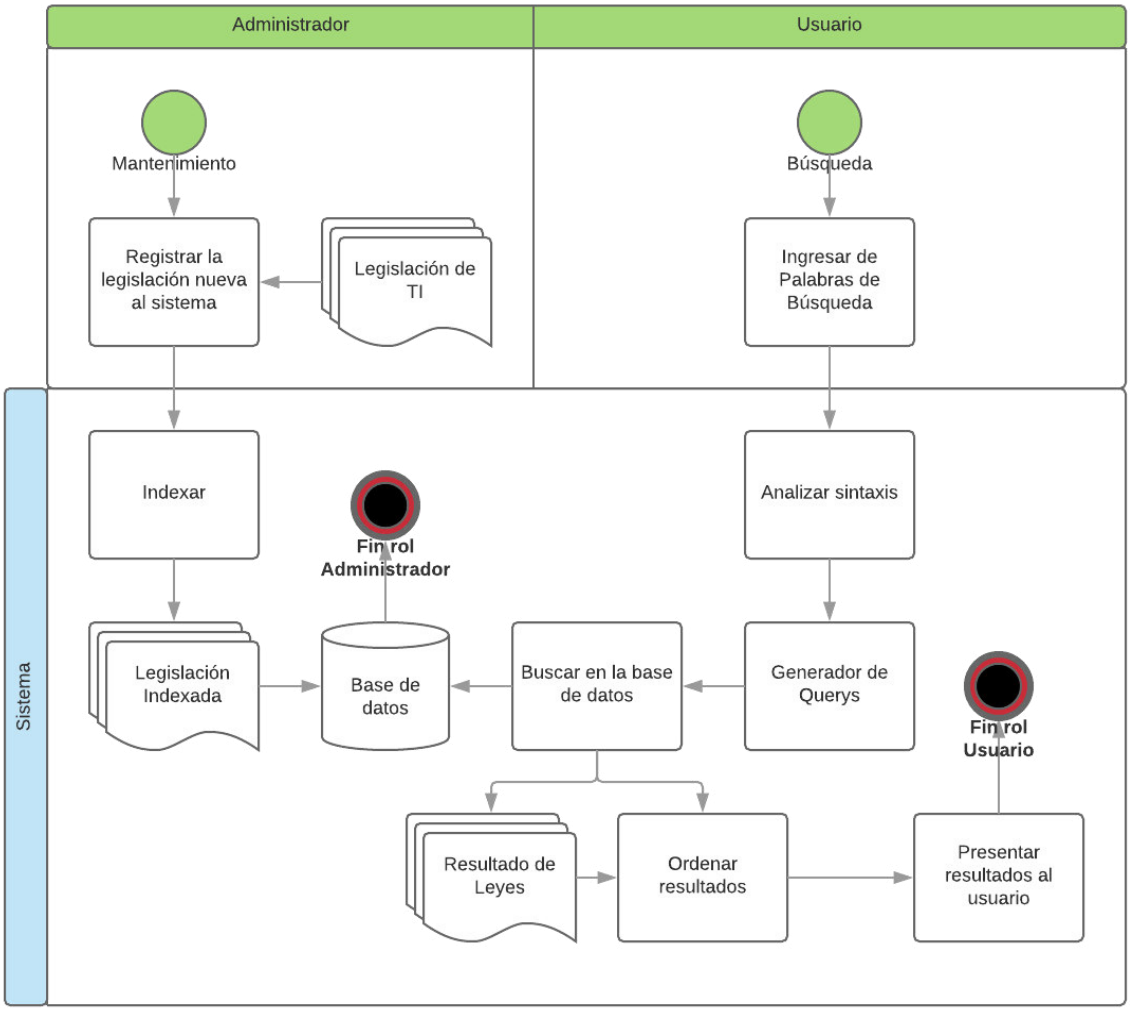


Figura 10. Diagrama de flujo del buscador propuesto

Dicho esto, en los siguientes subcapítulos se detallará acerca de las tareas y responsabilidades tanto del usuario común como del administrador, indicando una descripción del proceso que cada uno sigue. Luego, se verá minuciosamente los objetivos de cada uno de los procesos que realiza automáticamente el buscador, y además se dará a conocer el pseudocódigo o diagrama de flujo propio de cada uno de estos procesos, con la finalidad de mostrar o ejemplificar el desarrollo realizado.

4.3.3 Roles

Tal como se muestra en la figura 10, se puede distinguir que existen dos roles principales en el sistema: administrador y usuario. El administrador es aquel que gestiona el contenido de legislación que tendrá el sistema (mantenimiento), y el usuario es aquel que realiza las búsquedas de estos documentos en el sistema mediante consultas ingresadas.

4.3.3.1 Administrador

Respecto al administrador del sistema, cada vez que alguna nueva ley necesite ser añadida a la base de datos, el usuario administrador ingresará al sistema mediante sus credenciales de autenticación. Una vez aceptadas las credenciales, cargará la(s) ley(es) que corresponda(n) al sistema, el cual se encargará de almacenarlas en el servidor de archivos, para luego analizar estos documentos, indexarlos y guardar el índice para luego ser consultado. En caso se necesite editar o actualizar las características definidas de una documentación ya cargada, bastará con cargar nuevamente el documento con el mismo título con el que fue cargado el original, y, en esta nueva versión, establecer correctamente las características, pues simplemente se reemplazará el anterior.

4.3.3.2 Usuario

Por otro lado, el usuario ingresará las palabras claves para la consulta, las cuales serán verificadas por un analizador sintáctico que procesará la consulta, principalmente para traducir las siglas, obtener sinónimos y antónimos y luego, el generador de querys convertirá la salida del analizador sintáctico en una cadena que pueda entender el índice de forma óptima y priorizada. Finalmente, se realizará la consulta que obtendrá una lista de resultados, los mismos que, mediante un algoritmo de ordenamiento, serán presentados al usuario del más al menos relevante a través del propio navegador. Los procesos mencionados, tanto los que desencadena el usuario como los del administrador, serán detallados a continuación.

4.3.4 Procesos

Cada lenguaje en el mundo funciona diferente, y entender cómo funciona no es nada fácil, por lo que procesar tanto el texto de los documentos como la consulta da gran valor a un sistema, siendo esto lo que realmente permite aumentar la precisión de los resultados obtenidos con respecto a la consulta realizada. Todo buscador posee una serie de procesos internos que colaboran a su factor diferencial, ya que esto es lo que realmente hace que un buscador destaque sobre otros. En este caso, el factor diferencial está en esa serie de procesos establecidos para trabajar los textos en lenguaje español, particularmente en una terminología legal peruana, tanto en los documentos (durante la indexación), como en las consultas del usuario.

A continuación se procederá a detallar los procesos del flujo del sistema mencionados en el diagrama de flujo de la figura 10, a través del uso de pseudocódigos y diagramas de flujo, así como las razones de haber usado cierto método, estrategia o algoritmo.

4.3.4.1 Registrar los documentos

El administrador, al ingresar los documentos legislativos al sistema, podrá hacerlo en pdf o en otro formato que sea convertible a pdf. Deberá indicar solo las características básicas del archivo que está cargando, estas son las siguientes: la temática de la ley, la fecha de promulgación de la ley en cuestión, y rango de ley a la que pertenece. El sistema se encargará de indexarlo donde corresponda en el siguiente paso.

A continuación se muestra el pseudocódigo del proceso de registro de documentos (figura 11); y el diagrama de flujo correspondiente (figura 12).

```
1  Proceso RegistrarDocumentos
2      Leer documentos
3      Para Cada documento de documentos Hacer
4          Si  NO esPDF(doc) Entonces
5              documento <- convertirPDF(documento)
6          FinSi
7          Leer tema
8          agregar(tema,temas)
9          Leer fechaPromulgacion
10         agregar(fechaPromulgacion,fechas)
11         Leer rangoDeLey
12         agregar(rangoDeLey,rangos)
13     FinPara
14     indexar(documentos,temas,fechas,rangos)
15 FinProceso
```

Figura 11. Pseudocódigo del registro de documentos

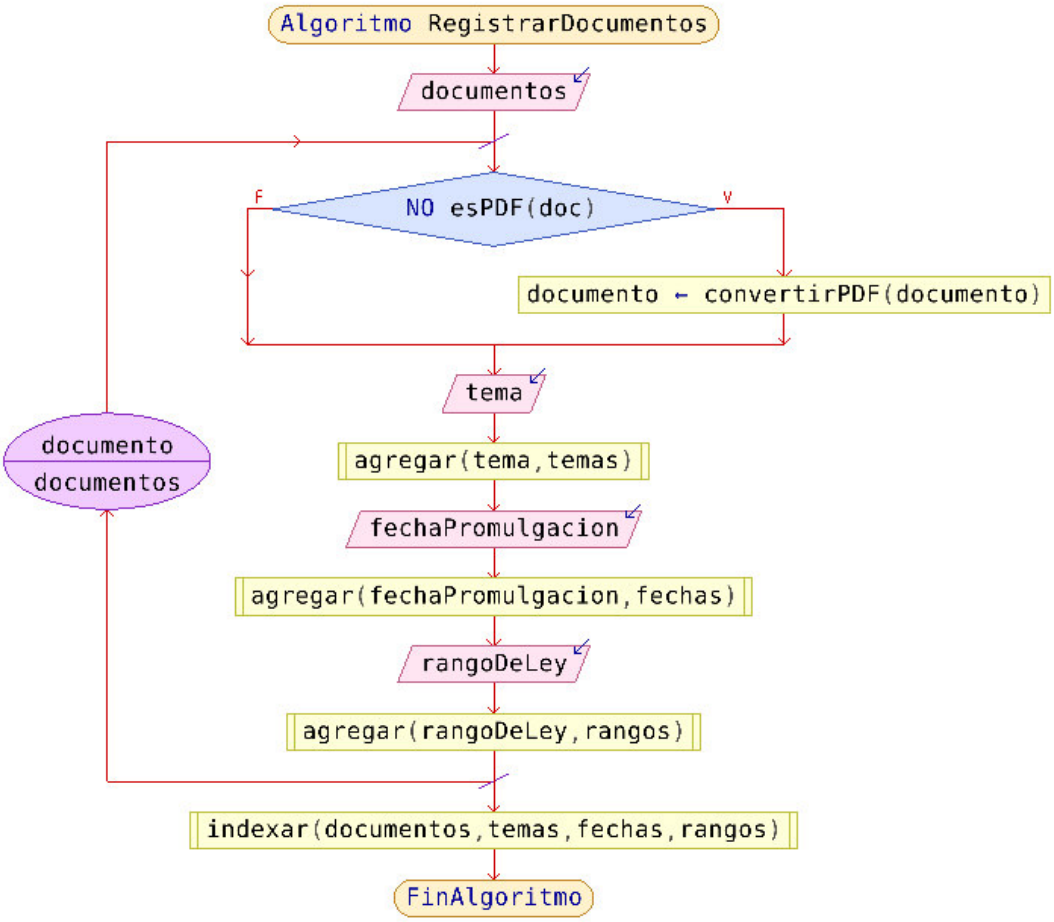


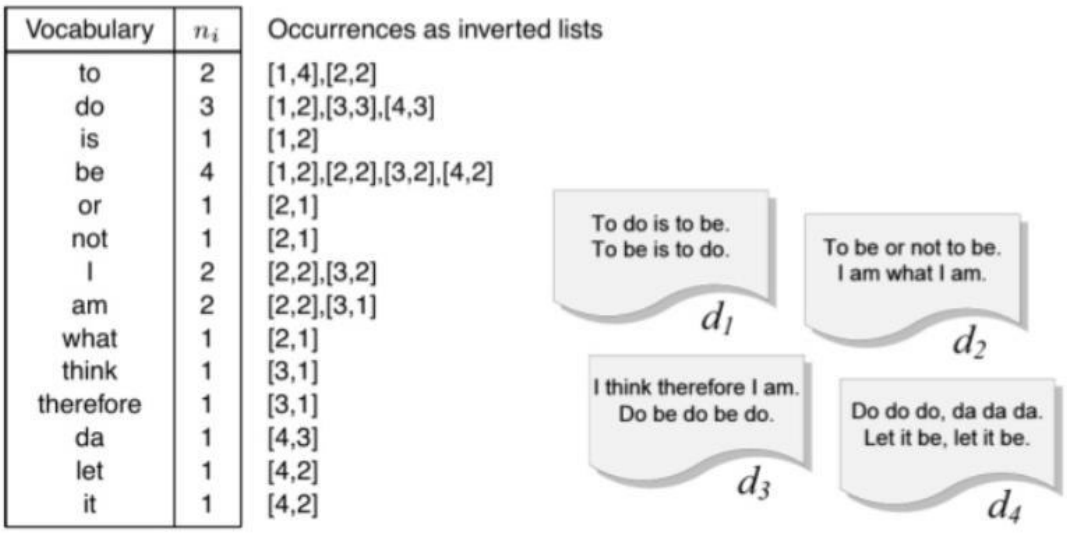
Figura 12. Diagrama de flujo del registro de documentos

4.3.4.2 Indexar los documentos

Cuando la cantidad de documentos o la cantidad de consultas es potencialmente grande (como en este caso), se debe crear un índice antes de proceder a la búsqueda. La indexación es el proceso realizado por el sistema, en el cual se establece una estructura de datos con la información del contenido de los documentos debidamente clasificados, y las ubicaciones de las palabras a indexar, todo esto mediante un criterio establecido durante la configuración del indexado, para que luego la búsqueda se realice a través de este índice. Por ejemplo, se puede configurar al índice para ignorar palabras sin contenido semántico, como los artículos, que no aportan a la búsqueda.

La indexación se puede hacer en diferentes bloques de texto por separado (resumen, texto completo, párrafos, palabras clave) por cada documento. Por otro lado, puede optarse por distintos tipos de estructura de datos, entre estos se tiene los siguientes: indexación con árbol de sufijos, indexación invertida, indexación por hipervínculos, entre otros.

En este caso, en la configuración de Solr, se define los siguientes grupos de texto o etiquetas para el indexado: título, contenido, fecha de promulgación, temática y jerarquía, y se eligió la indexación invertida, pues es la más recomendada para las búsquedas a texto completo (Lashkaria, Ensanb, Bagheric, & Ghorbani, 2017) (Konow & Navarro, 2012) (Bast & Weber, 2006), como sucede en esta propuesta. La indexación invertida almacena la referencia de la ubicación de una palabra en los documentos en los que aparece, para así encontrar fácilmente los documentos en los que se encuentra la palabra que se busca. En la figura 13, se muestra un ejemplo práctico de la indexación invertida.



Vocabulary	n_i	Occurrences as inverted lists
to	2	[1,4],[2,2]
do	3	[1,2],[3,3],[4,3]
is	1	[1,2]
be	4	[1,2],[2,2],[3,2],[4,2]
or	1	[2,1]
not	1	[2,1]
I	2	[2,2],[3,2]
am	2	[2,2],[3,1]
what	1	[2,1]
think	1	[3,1]
therefore	1	[3,1]
da	1	[4,3]
let	1	[4,2]
it	1	[4,2]

Figura 13. Ejemplo de indexación invertida

Fuente: Slideshare

Siguiendo con la figura 13, se observa que d1, d2, d3, y d4 conformarían el universo de documentos que fueron indexados (en este caso la legislación peruana de TI); el índice, al lado izquierdo, muestra el listado de distintas palabras encontradas en los documentos y el número de documentos en los que aparece. Por último, el resultado de una consulta de cualquiera de las palabras se muestra en medio en pares ordenados, siendo el primer término el documento en que se encuentra, y el segundo término, la cantidad de veces que se encuentra en dicho documento.

Si se hace un ejercicio mental rápido, se puede ver que con dicho índice (cuadro de la figura 13) es fácil responder cuántas veces aparece la palabra “be”, y en qué documentos, por lo que, haciendo el símil, el tiempo de respuesta de este tipo de índices es corto. Sin embargo, si de igual modo se hace el ejercicio de realizar desde cero el índice, se notará que tomará tiempo, pues se tiene que leer todos los documentos y contar las apariciones. Precisamente esta es la desventaja de la indexación invertida: el tiempo de indexación es alto comparado con otro tipo de indexados.

Por ello, se decidió que el buscador almacene en segundo plano la información de las cargas que se realicen, para hacer efectivo el indexado en periodos donde los usuarios no estén realizando consultas para no afectar los tiempos de respuesta.

A continuación se muestra el pseudocódigo del proceso de indexado (figura 14) y el diagrama de flujo correspondiente (figura 15).

```
Proceso indexar
    leer documentos, temas, fechas, rangos
    Uso <- CalcularUsoCPUyRAM()
    Si Uso > UsoPromedio Entonces
        programarIndexado(documentos,temas,fechas,rangos,datetime)
    SiNo
        Para Cada documento de documentos Hacer
            id <- generarId(documento)
            titulo <- obtenerTitulo(documento)
            IndiceIDs(id,documento)
            IndiceTitulo(titulo,documento)
            IndiceTema(tema,documento)
            IndiceFecha(fecha,documento)
            IndiceRango(rango,documento)
            contenido <- obtenerTexto(documento)
            Para Cada palabra de contenido Hacer
                Si ExisteEnElIndice(palabra) Entonces
                    IndiceContenido(palabra,documento)
                SiNo
                    AgregarAlIndice(palabra)
                    IndiceContenido(palabra,documento)
                FinSi
            FinPara
        FinPara
    Fin Si
FinProceso
```

Figura 14. Pseudocódigo del proceso de indexación

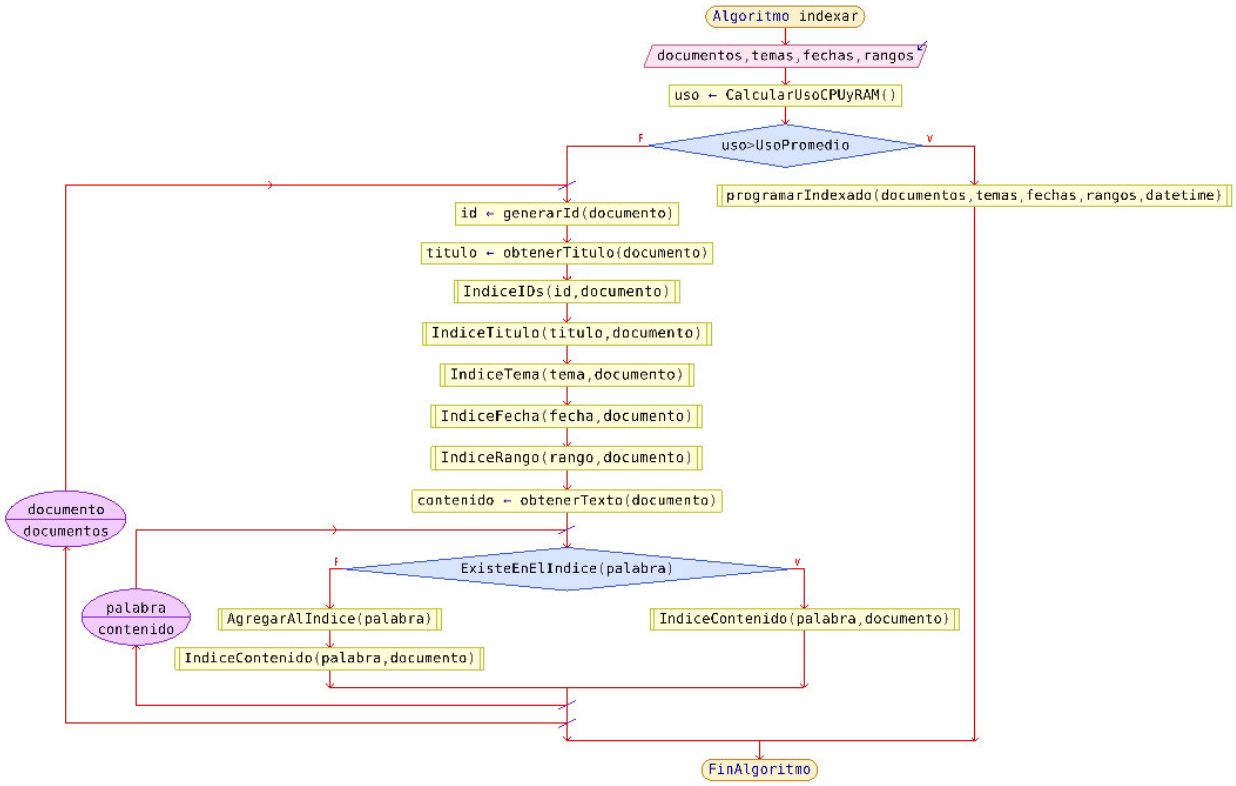


Figura 15. Diagrama de flujo del proceso de indexación

4.3.4.3 Analizar sintaxis y semántica

Este proceso la realiza el sistema después de que el usuario ingresa una frase de búsqueda con el fin de realizar una consulta. Su labor es analizar y procesar las palabras ingresadas y transformarlas de modo que no solo se busque explícitamente lo que el usuario ha ingresado, sino que también se buscarán las palabras ignorando el género y el número, los sinónimos de las palabras ingresadas, y la traducción larga de las siglas consultadas (en el caso de que se hayan ingresado).

Otro de los procesos que tiene este componente es la corrección de errores gramaticales, el reemplazo de palabras por antónimos, la eliminación de espacios en blanco o palabras sin contenido semántico como los artículos y remover signos de puntuación.

A continuación se muestra el pseudocódigo del proceso de análisis de sintaxis (figura 16) y el diagrama de flujo correspondiente (figura 17).

```
Proceso AnalizarSintaxisYSemantica
    Leer fraseDeBusqueda
    eliminarEspacios(fraseDeBusqueda)
    eliminarConectores(fraseDeBusqueda)
    eliminarSignos(fraseDeBusqueda)
    corregirErrores(fraseDeBusqueda)
    query <- ''
    Para Cada palabra de fraseDeBusqueda Hacer
        query <- query + ObtenerRaiz(palabra)
        query <- query + ObtenerSinonimos(palabra)
        query <- query + ObtenerAntonimos(palabra)
        query <- query + ObtenerSiglas(palabra)
    FinPara
FinProceso
```

Figura 16. Pseudocódigo del proceso de analisis sintactico y semántico

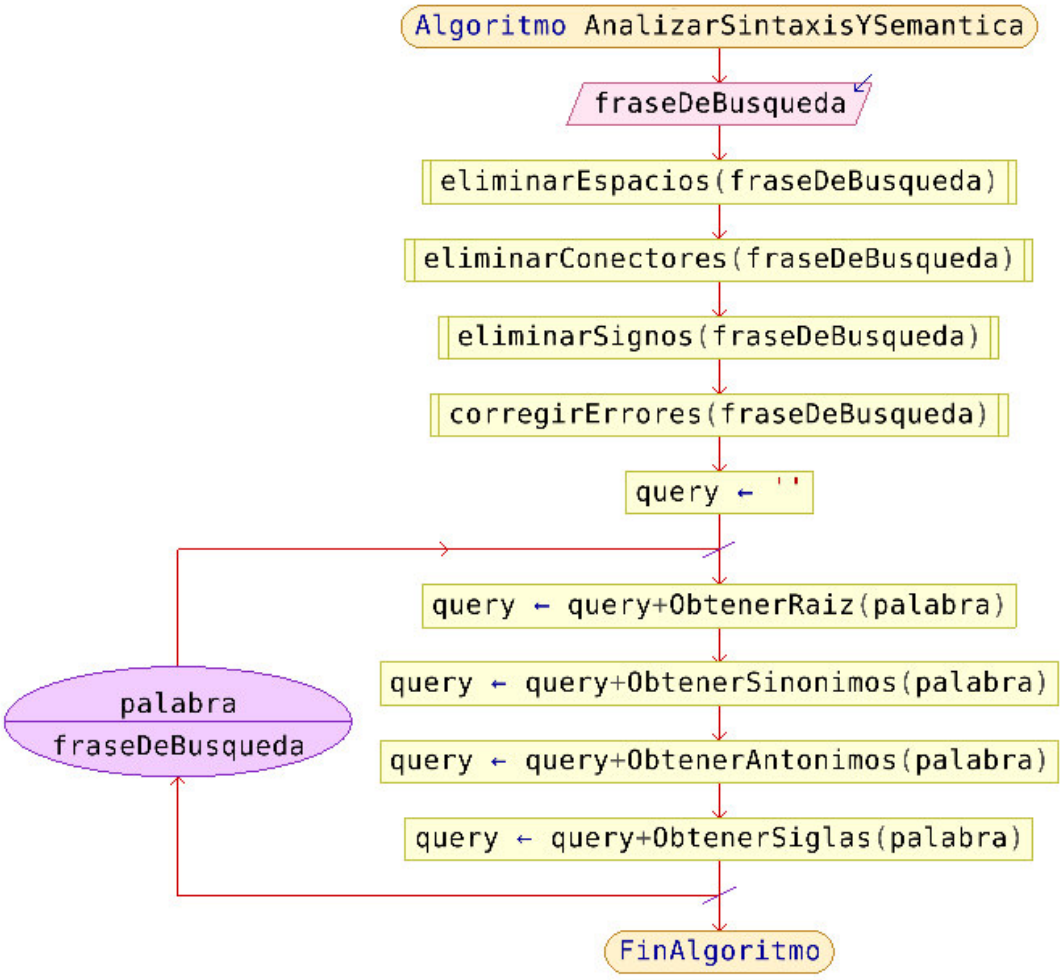


Figura 17. Diagrama de Flujo del análisis sintáctico y semántico

4.3.4.4 Generar query

Con las palabras ingresadas por el usuario debidamente procesadas en el paso anterior, se procede a formar las consultas para realizar en el servidor. En el anterior proceso, inevitablemente se agrega más palabras para ser consultadas en el índice. Pero la idea no es solamente hacer una consulta con más palabras para obtener un mayor resultado, sino, darles una prioridad a cada una, y eso es lo que se realiza en este proceso. Por ejemplo, si el usuario busca “corredora”, se le da una mayor sensación de precisión al usuario si se muestra primero las coincidencias exactas de lo

que ha buscado, para luego, en segundo lugar, mostrar las coincidencias con la palabra “corredor” (ignorando el género), y, en tercer lugar, mostrar las coincidencias con la raíz “corr-”.

También hace falta que prioritariamente la consulta se realice sobre el título, en segundo lugar sobre el contenido y en tercer lugar en la temática, con el objetivo de obtener un mayor grado de precisión. Además, como valor agregado, se puede implementar el uso de valores lógicos, de modo que el usuario pueda establecer condiciones durante su consulta, pudiendo así incluir, excluir o forzar términos de su cadena de búsqueda.

A continuación se muestra el pseudocódigo del proceso de generación de querys (figura 18) y el diagrama de flujo correspondiente (figura 19).

```
1  Proceso GenerarQuery
2      Si HayValoresLogicos(query) Entonces
3          EstablecerValorLogico(query)
4      FinSi
5      EstablecerPrioridades(query)
6      RealizarBusqueda(query)
7  FinProceso
```

Figura 18. Pseudocódigo del proceso de generación de querys

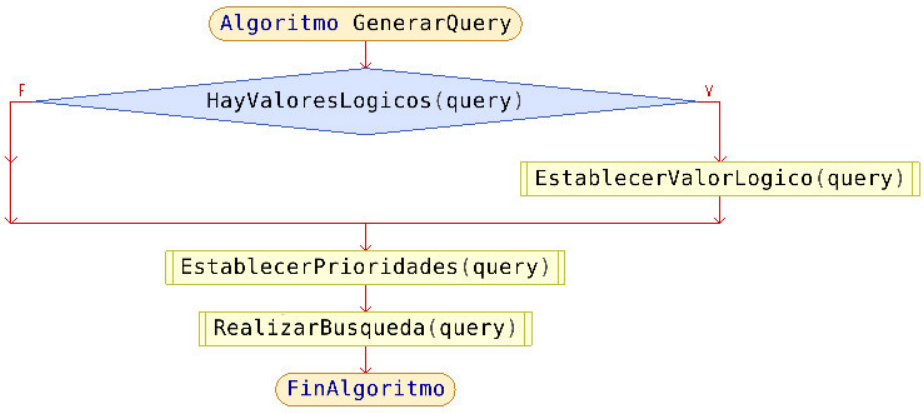


Figura 19. Diagrama de flujo del proceso de generación de querys

4.3.4.5 Buscar en la base de datos

Con la consulta ya formada a partir de las palabras ingresadas por el usuario, siguiendo los criterios establecidos en el proceso anterior, se procederá a realizar la búsqueda en el índice de la documentación. Como se vio en el ejemplo de la figura 13, la consulta se hace sobre el índice, y este trae como resultado la referencia al documento el cual contiene la palabra buscada, pero también se puede obtener su ubicación en el mismo. Esto permitirá mostrarle al usuario un extracto del documento en el cual aparece la palabra consultada, a modo de vista previa, tal como lo hace Google al mostrar un breve texto del contenido de una web en donde aparece la consulta que se ha realizado.

Entonces, se obtiene un conjunto de documentos que correspondan a la consulta generada; pero como es lógico y como se vio en el ejemplo de la figura 13, algunos documentos tendrán una mayor coincidencia que otros, por lo que hará falta que sean ordenados (en el siguiente proceso).

A continuación se muestra el pseudocódigo del proceso de búsqueda en el índice (figura 20), la base de datos y el diagrama de flujo correspondiente (figura 21).

```
1  Proceso RealizarBusqueda
2      coincidenciaGlobal = 0
3      Para Cada token de queryProcesada Hacer
4          Para Cada documento de documentos Hacer
5              coincidencias <- coincidencias + coincidencia(token, indice, documento)
6              n <- cantidad(coincidencias)
7              vista<-vistaPrevia(token, indice, documento)
8              coincidenciaGlobal <- coincidenciaGlobal + n
9          FinPara
10     Fin Para
11 FinProceso
```

Figura 20. Pseudocódigo del proceso de búsqueda

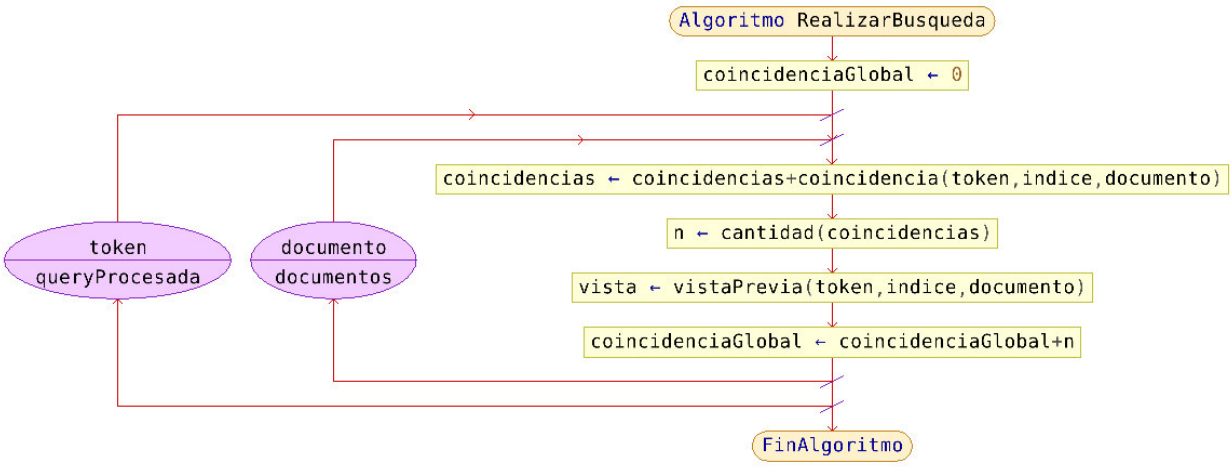


Figura 21. Diagrama de flujo del proceso de búsqueda

4.3.4.6 Ordenar resultados

Aquí se establece el orden en que los documentos encontrados aparecerán al usuario. Hay varias maneras de hacerlo; algunos optan por evitar la redundancia, mostrando solo un documento por tema (Guo, Zhang, & Kuang, 2016), mientras que otros eligen mostrar los documentos en los que aparece más veces las palabras claves ingresadas; sin importar si los N primeros documentos tienen un contenido muy similar.

Atrás de esta lógica hay cálculos matemáticos: la puntuación que se le da a un documento según la consulta responde a un algoritmo de similaridad que hay que establecer según la conveniencia para el buscador a desarrollar. Como ya se describió en la historia de usuario HU12, Solr tiene un componente de puntuación el cual consiste básicamente en la multiplicación del valor representado por la cantidad de apariciones de un término en un documento entre la cantidad de palabras de ese documento, dividido por la cantidad de apariciones de ese mismo término en todo el conjunto de documentos entre la cantidad de palabras de todo el conjunto.

A continuación, se muestra el pseudocódigo del proceso de ordenamiento de resultados (figura 22) y el diagrama de flujo correspondiente (figura 23).

```
1   Proceso OrdenamientoDeResultados
2       Para Cada documento de documentos Hacer
3           palabras <- cantidadPalabras(documento) // palabras indexables
4           numCoincidencias <- obtenerCoincidencias(documento)
5           palabrasTotal <- cantidadPalabrasConjunto(documentos)
6           score <- (numCoincidencias/palabras) / (coincidenciaGlobal/palabrasTotal)
7           scores <- scores + score
8       FinPara
9       ordenarDesc(scores)
10      presentarResultados(scores, documentos, vistaPrevia)
11  FinProceso
```

Figura 22. Pseudocódigo del ordenamiento de resultados

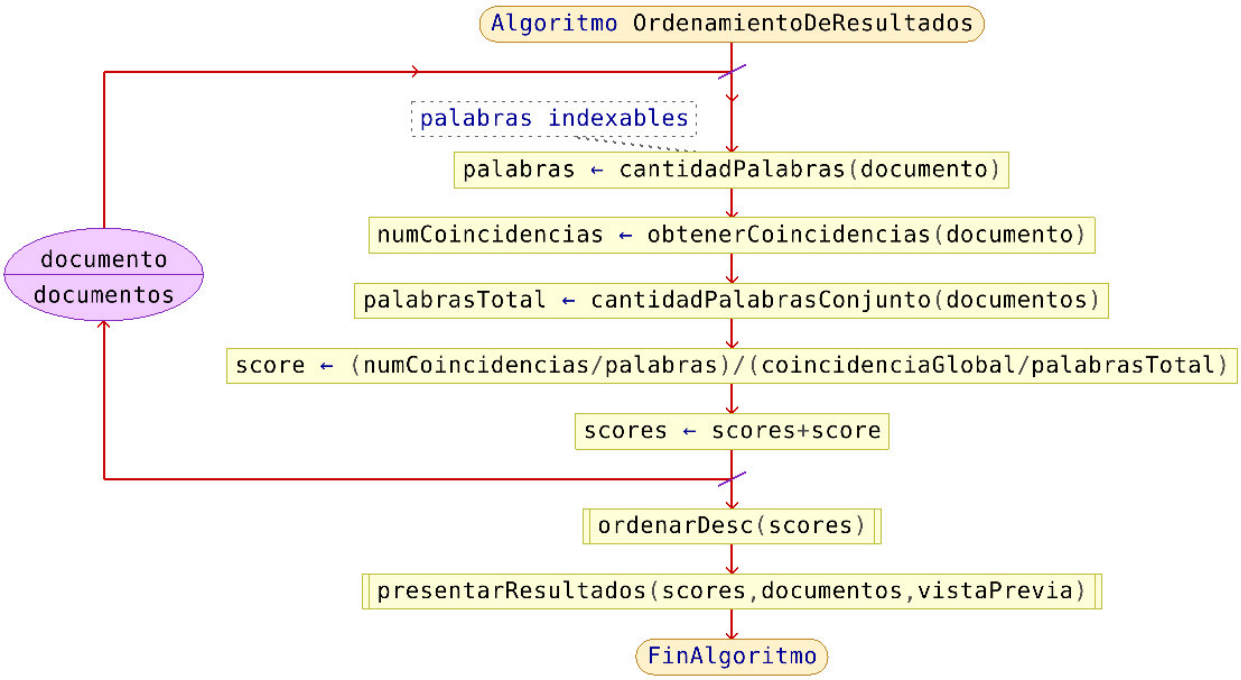


Figura 23. Diagrama de flujo del ordenamiento de resultados

4.3.4.7 Presentar resultados

La forma en la que se presentarán los resultados tiene que ser, como se planificó en el análisis de requerimientos, de una forma simple. Por ello, para realizar el diseño de las interfaces del sistema, se siguieron los lineamientos y buenas prácticas que utilizan otros buscadores como Google o Bing, con la finalidad de que el usuario aprenda a usar la herramienta intuitivamente, colaborando con esto a que la percepción del usuario sea buena, generando aceptación por la herramienta. El diseño de nuestra propuesta para la interfaz principal de la búsqueda se puede observar en la figura 24.

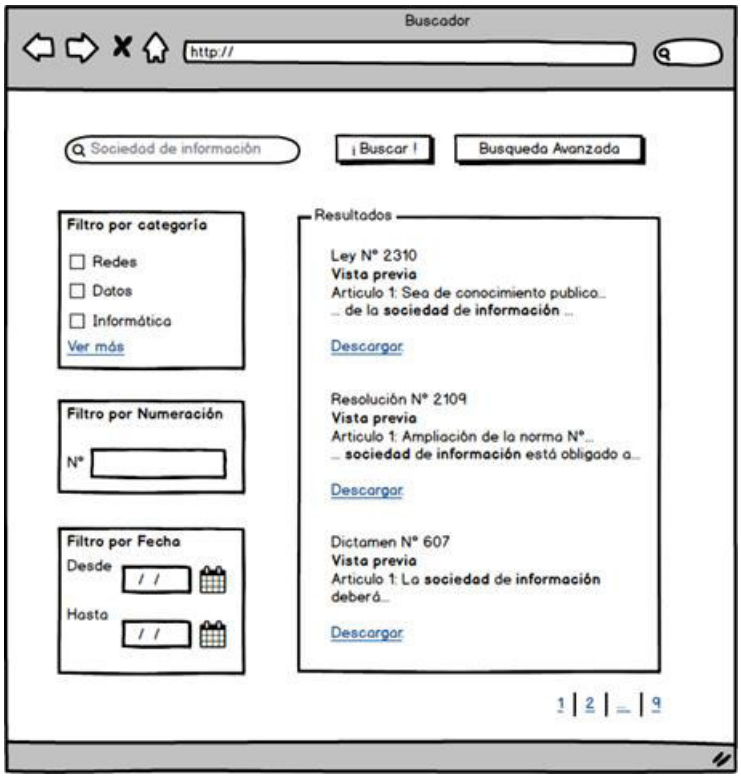


Figura 24. Diseño de interfaz principal del usuario

Debe recordarse que en el anterior proceso se obtiene la vista previa de documentos que tienen alguna coincidencia, además de traerlos en orden descendente, según la puntuación obtenida, siendo esta puntuación brindada por la precisión del documento respecto a la consulta.

4.3.5 Interfaz gráfica

A continuación, en las figuras 25, 26, 27 y 28, se presenta las interfaces gráficas del buscador, siguiendo el maquetado mostrado en la figura 24, ya que este fue aprobado por el cliente Carlos Ferreyros.

Figura 25. Diseño final de interfaz gráfica de usuario del buscador: Página principal

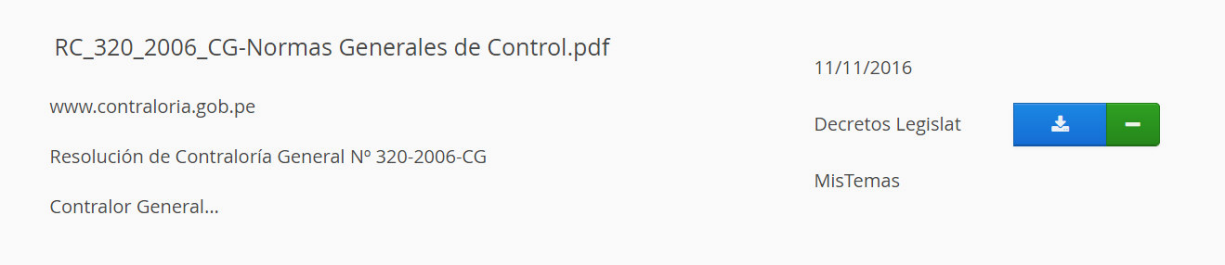


Figura 26. Ejemplo de Resultado: Titulo, resumen, características, botón de descarga

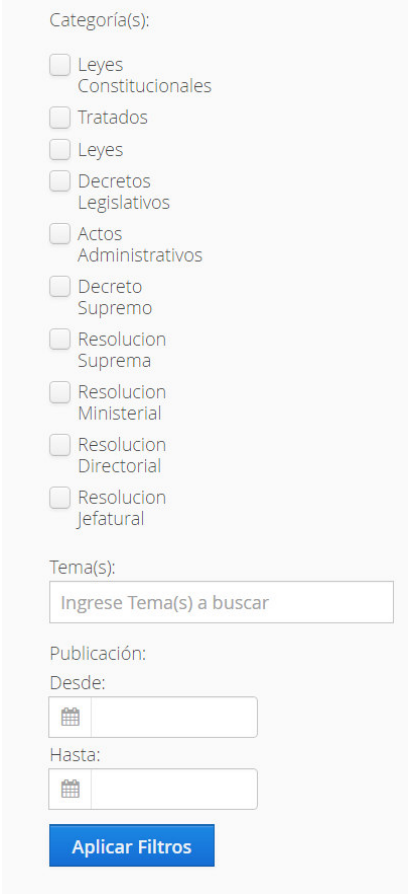


Figura 27. Diseño del panel de filtros

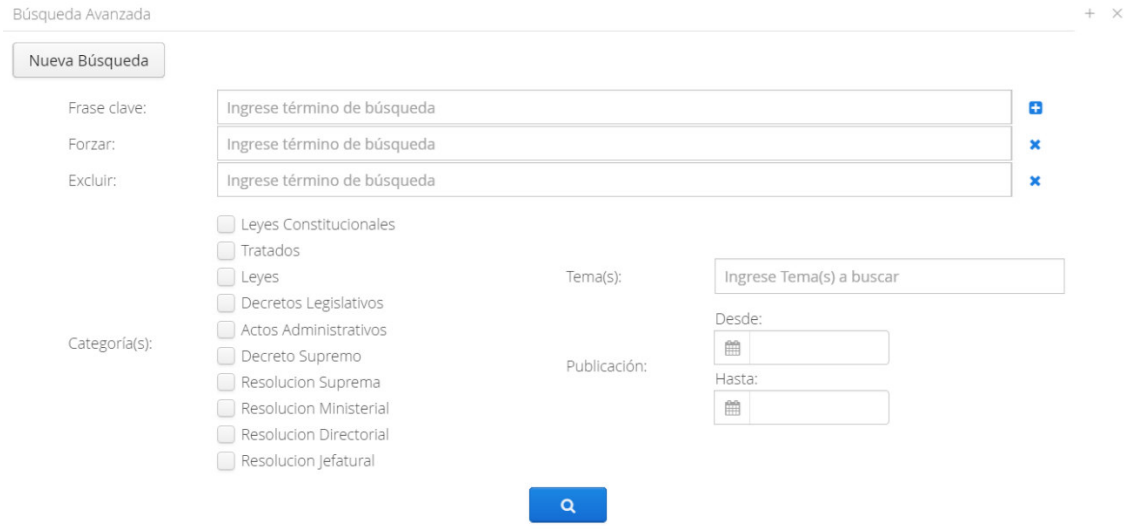


Figura 28. Diseño de la interfaz de búsqueda avanzada

4.3.6 Experimento: Ejemplo funcional

A continuación se muestra DoLaw en funcionamiento, ingresando consultas que puedan probar las funcionalidades, para luego desplegarlos y validarlos en el siguiente capítulo, y sea el usuario final quien realmente valide el buscador desarrollado.

4.3.5.1 Definir consultas para el experimento

A modo de demostración, se mostrará los resultados obtenidos para tres consultas. Las consultas elegidas serán:

a) firma digital (consulta simple)

b) dni (siglas)

c) la auditoría tecnica de entidadas (palabra con tilde, palabra con tilde omitida, conectores, error ortográfico)

4.3.5.2 Resultados del experimento

A continuación se muestran los resultados de las tres consultas que fueron parte del experimento.

a) firma digital

Con este ejemplo se muestra en la figura 29 que el resultado obtiene resultados singulares y plurales de cada una de las palabras ingresadas, por lo que se comprueba el funcionamiento correcto de la eliminación de afijos

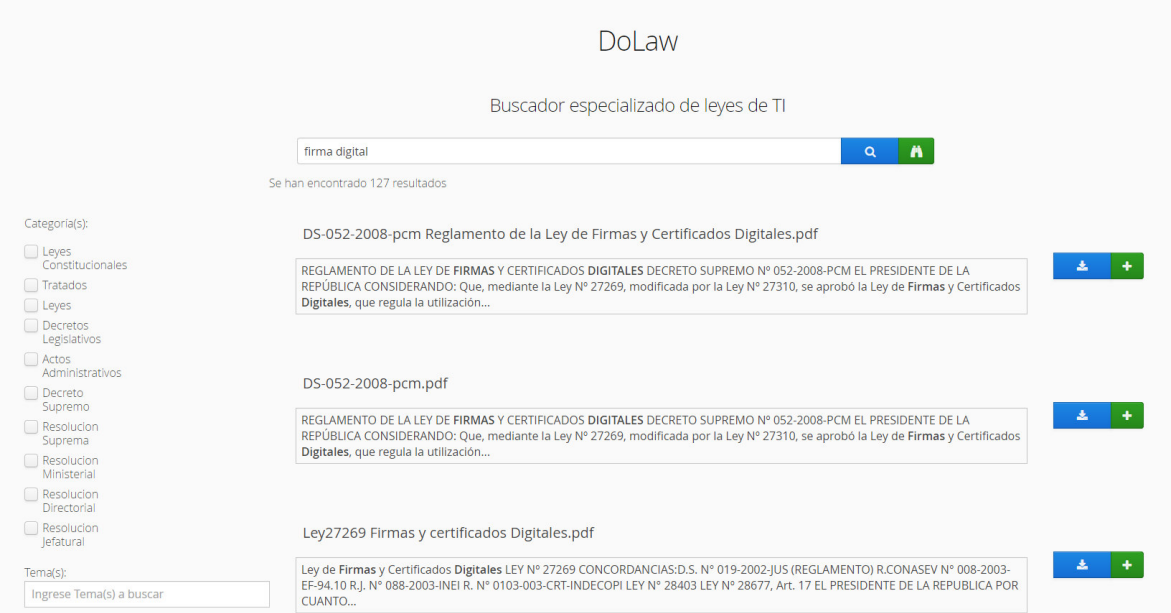


Figura 29. Resultados de query “firma digital”

b) dni

Con este ejemplo se muestra en la figura 30 el funcionamiento de la interpretación de siglas peruanas, buscando “documento nacional de identidad” y “dni” en paralelo.

Figura 30. Resultados de query “dni”

c) la auditoría tecnica de entidadas

Con este ejemplo se muestra en la figura 31 la correcta eliminación de conectores, pues no son resaltados en los resultados, además, encuentra resultados sin tilde para las palabras con tilde y visceversa. El buscador también entiende el error ortográfico en “entidadas”, encontrando “entidades” y su versión singular, “entidad”.

Figura 31. Resultados de query “la auditoría tecnica de entidadas”

4.3.5.3 Precisión y exhaustividad del experimento

Para calcular la precisión y exhaustividad, se debe tener la cantidad de resultados obtenidos, la cantidad de resultados relevantes y la cantidad de resultados omitidos. El Dr. Ferreyros, quien nos facilitó el acceso a los documentos de legislación, fue el experto en legislación peruana de tecnología de información que identificó si un resultado obtenido por el buscador relevante para la consulta, y si algún resultado fue omitido, mientras que el sistema brinda en cada consulta la cantidad de resultados obtenidos. Recordemos como se calculan los valores de precisión y exhaustividad.

$$Precisión = \frac{|\{documentos\ relevantes\} \cap \{documentos\ recuperados\}|}{|\{documentos\ recuperados\}|}$$

$$Exhaustividad = \frac{|\{documentos\ relevantes\} \cap \{documentos\ recuperados\}|}{|\{documentos\ relevantes\}|}$$

En los experimentos, los valores de precisión y exhaustividad son los mostrados en la tabla 16

Tabla 16. Calculo de precision y exhaustividad de experimentos de uso de DoLaw

Query	Resultados obtenidos	Resultados relevantes	Precisión	Exhaustividad
firma digital	127	113	113/127 = 88.98%	113/113 = 100%
dni	187	105	105/187 = 56.15%	105/105 = 100%
la auditoría tecnica de entidadas	178	178	178/178 = 100%	178/178 = 100%

En promedio, la precisión del sistema según los experimentos realizados es de 87.06%; y la exhaustividad es del 100%.

CAPÍTULO 5: VALIDACIÓN

En este capitulo se validará el sistema propuesto a través de la satisfacción del usuario final (Liu & Guo, 2008) (Rana, Dwivedi, Williams, & Weerakkody, 2015). Se realizará mediante encuestas a diecisiete (17) usuarios a quienes les brindamos acceso al buscador por diez (10) días. Se menciona la configuración y condiciones en las que se brindó el sistema, así mismo, las métricas de evaluación y los resultados obtenidos.

5.1 Diseño de la validación

Para validar que DoLaw supera a los buscadores de leyes ya existentes, se realizaron encuestas en el estudio jurídico Ferreyros & Ferreyros y en el consultorio jurídico Sevilla Parrilla, así, se vieron algunos de muchos posibles casos donde se puede implementar este buscador.

En primer lugar, se identificó que ambos estudios utilizaban el Sistema Peruano de Información Jurídica (SPIJ) como buscador de legislación. Al ver esta coincidencia, se preguntó al Dr. Ferreyros durante una entrevista, si SPIJ era el sistema más usado y por qué, a lo que respondió que sí, pues actualmente SPIJ era el sistema con mayor alcance, debido a que nació mediante una inversión del estado peruano, pero que aún tenía mucho por mejorar. En consecuencia, se decidió hacer la comparación únicamente con este sistema.

Luego se establecen las métricas, y, después de establecer la situación actual del usuario, se le brinda la herramienta, para luego evaluar la satisfacción que le produce el uso de DoLaw con respecto a SPIJ, para finalmente generar los resultados comparativos entre ambos sistemas.

Dichos pasos se muestran en la figura 32.

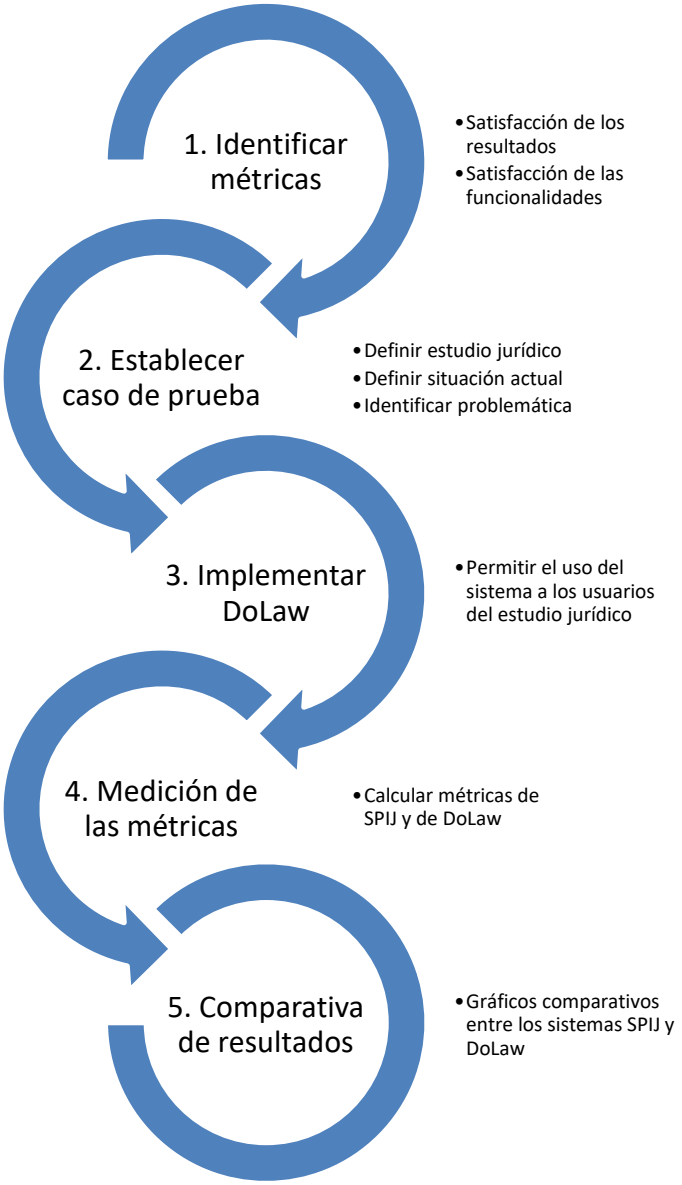


Figura 32: Diseño de la validación

5.2 Métricas

Para establecer las métricas por las que se comparará DoLaw con SPIJ, se decidió que, debido a no conocer las características del servidor de SPIJ, se evaluará las métricas que el usuario pueda brindar en igualdad de condiciones, estas son las siguientes: satisfacción de resultados obtenidos

en una consulta (definido por la sensación de precisión y exhaustividad que tenga el usuario) y satisfacción de funcionalidades adicionales.

- Satisfacción de los resultados: En esta métrica, se pregunta a los usuarios si consideran que los documentos obtenidos por el sistema son los resultados que esperaban (relevantes), en exactitud (precisión) y en cantidad (exhaustividad), así, se le da la opción de responder en una escala lineal del 1 al 5, siendo 1 “mucho peor” y 5 “muy mejor”. Dicha escala establecida para encuestas es conocida como Escala de Linkert.

- Satisfacción de las funcionalidades: En esta métrica, se consulta a los usuarios si consideran que tanto los filtros, el uso de sinónimos, el uso de siglas, búsqueda por raíces, entre otros, incluyendo el diseño de DoLaw, son aportes que funcionan como deberían y si son de utilidad. También se le brinda la opción de responder bajo la misma escala lineal del 1 al 5.

5.3 Configuración del sistema

Se presentó DoLaw para ambos estudios, brindándole acceso libre por cinco (5) días para que así tengan la posibilidad de familiarizarse con la herramienta y puedan hacer pruebas de su funcionamiento libremente. Para la demostración, se consiguieron 317 leyes válidas de distinto rango de ley, distinto tema y fecha de publicación (todas del 2010 en adelante), las cuales representaban una muestra suficientemente amplia para la evaluación, según el Dr. Ferreyros. La aplicación fue desplegada y ejecutada desde una computadora con arquitectura:

- Procesador: Intel Core i7-4720HQ – CPU 2.60GHz
- RAM: 16GB
- Sistema operativo: Windows 10 de 64 bits.

5.4 Encuestas

Para medir la satisfacción del usuario, se procedió a realizar encuestas, con la finalidad de medir la percepción actual de SPIJ comparado con DoLaw, tanto en los resultados obtenidos como en las funcionalidades (ya que estas fueron nuestras métricas). Las encuestas eran breves, pues, como se mencionó, consistían en respoder solo cinco preguntas de escala lineal, lo que permitirá evaluar las métricas establecidas.

5.4.1 Población

La población a la que se impartieron las encuestas se distribuye tal como se muestra a continuación en la tabla 17.

Tabla 17. Distribución de la población encuestada

Edad \ Sector	Profesionales en Legislación	Profesionales de TI
< 25 años	3	1
25-40 años	7	1
> 40 años	4	1

Como se ve, se decidió tener una muestra variada y representativa de los potenciales principales usuarios de DoLaw, permitiendo que existan usuarios con estudios o trabajos en el sector legislativo, y otro grupo más pequeño en el sector de tecnologías de información. Las edades de los mismos también son variadas, y se decidió así porque habría que tener en cuenta que las personas de menor edad son más propensas al cambio y a aprender el uso de nuevas herramientas con mayor facilidad.

5.4.2. Ejecución de la encuesta

Las preguntas fueron seleccionadas con el fin de cubrir las métricas establecidas. Se utilizó Google Forms para diseñar las encuentas debido a la facilidad que le brinda al usuario para responder. Las preguntas pueden ser encontradas en el Anexo B-1, mientras que las respuestas a dichas preguntas (las cuales fueron contestadas después que los usuarios probaron el sistema) están adjuntas en el Anexo B-2.

5.5 Resultados

Los resultados de la encuesta realizada han sido representados en los cuadros presentados en las figuras 33, 34 y 35 en forma comparativa entre ambos sistemas.

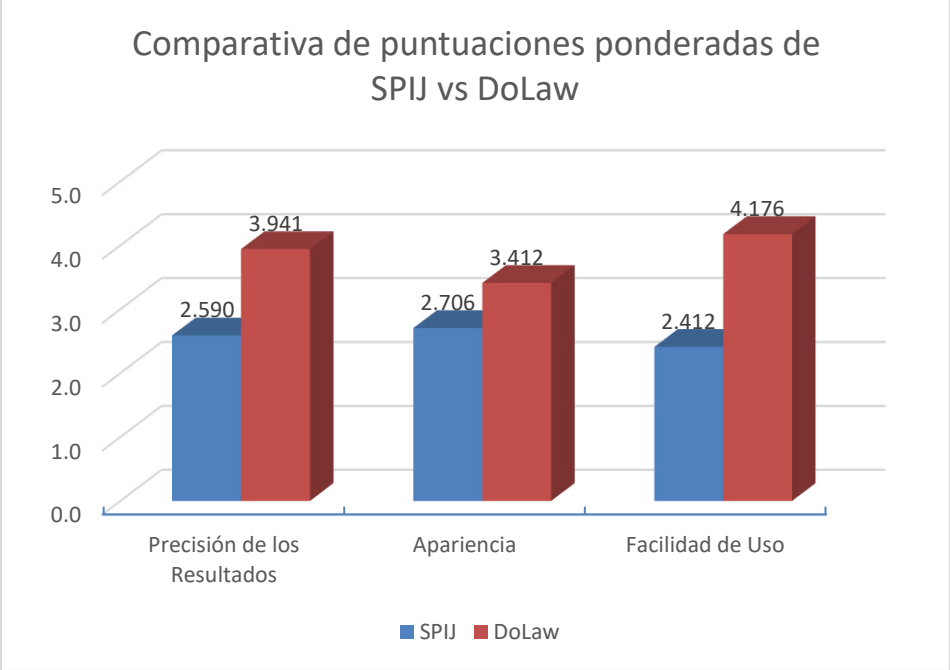


Figura 33. Comparativa de puntuaciones ponderadas obtenidas por SPIJ y DoLaw

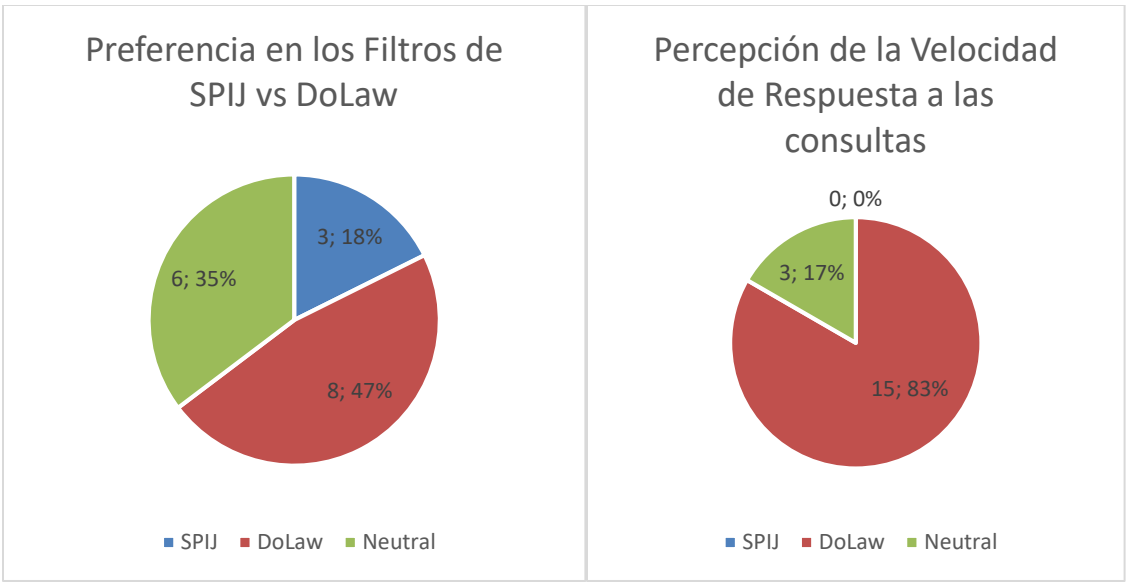


Figura 34. Comparativa de percepción de la utilidad de los filtros y la sensación de velocidad

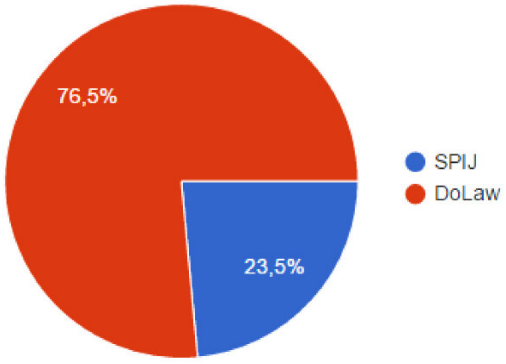


Figura 35. Preferencia general de un DoLaw sobre SPIJ (en caso tuvieran el mismo contenido)

Se observa que, en promedio, las métricas indican que la opinión de los usuarios encuestados acerca de DoLaw es mejor en las áreas que se plantearon mejorar (funcionalidades, resultados obtenidos, precisión, exhaustividad, e incluso apariencia y facilidad de uso), así como también la preferencia general de los sistemas muestra mayor cantidad de votos por DoLaw, poniendo como premisa el supuesto de que ambos tuviesen el mismo contenido de legislación.

CAPÍTULO 6: CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se menciona las conclusiones del proyecto realizado, así como las limitaciones que tiene el mismo. Finalmente, se dará sugerencias para futuros trabajos enfocados en la mejora o en el desarrollo de proyectos similares.

6.1 Conclusiones

6.1.1 Conclusión General

Se desarrolló con éxito un buscador especializado de legislación peruana de TI que cumplió con las necesidades que tiene el mercado al solucionar la problemática existente, incluyendo funcionalidades de dominio específico y mejorando los resultados obtenidos; y con todo ello, la experiencia de usuario tal como fue planteado. Además, el buscador DoLaw posee las funcionalidades que fueron planteadas en el objetivo general de la tesis, es decir, incluyó algoritmos de búsqueda a texto completo, algoritmos de búsqueda semántica, interpretación de consultas y también configuraciones especializadas que permitieron el filtrado de resultados, según sea solicitado por el usuario.

6.1.2 Conclusiones Específicas

6.1.2.1 Literatura

Después de la revisión del estado del arte, ya se conoce algunos algoritmos de soporte para las búsquedas como, por ejemplo, algoritmos de ordenamiento de resultados, de eliminación de redundancia, de agrupación de documentos por temática mediante algoritmos genéticos, entre

otros, los cuales fueron evaluados para ver si era necesario aplicarlos en DoLaw, según las necesidades que presentaba el área de aplicación (legislación), siguiendo las mejores prácticas.

La literatura también permitió conocer que para evaluar el desempeño del sistema, las métricas usadas son la precisión y la exhaustividad de los resultados, según la cadena de búsqueda; así como la percepción de los usuarios que al fin y al cabo serán quienes utilicen el sistema.

6.1.2.2 Arquitectura del sistema

Se diseñó una arquitectura de tal modo que el sistema pueda ser accedido desde cualquier dispositivo con un navegador, usando un servidor Solr donde se hace y almacena el índice, un servidor de archivos donde se almacenan todos los documentos, y un servidor de aplicaciones web, donde estará desplegado el buscador.

6.1.2.3 Desarrollo del sistema

Se desarrolló el sistema con los algoritmos planteados. El algoritmo de búsqueda a texto completo permitió que se busquen todas las palabras del contenido de todos los documentos, aumentando así la precisión. El algoritmo de búsqueda semántica y el de interpretación de consultas permitieron la inclusión de diccionarios de siglas, de sinónimos y de antónimos, con lo que se logró aumentar la exhaustividad al buscar no solo las palabras textuales que ingresa el usuario, sino sus principales equivalencias. También, al generar la consulta, se establecía una prioridad mayor a las palabras ingresadas por el usuario con respecto a las sugerencias, ayudando así a aumentar los índices de precisión y exhaustividad. Por último, las configuraciones especializadas permitieron filtrar los resultados según sea solicitado por el usuario.

Además, se diseñó una interfaz simple tanto en diseño, en colores, así como en cantidad de botones y opciones en la pantalla principal, y así ofrecer al usuario directamente lo que quiere:

realizar una búsqueda. Esto se realizó con base en los estándares establecidos por empresas que han desarrollado herramientas de búsqueda como Google o Bing.

6.1.2.4 Validación

Se consiguió 317 leyes de tecnología de información, de categorías difententes, todas en el período de entre el 2000 y el 2018. Además, se estableció una población de encuestados de 17 personas, tanto abogados como ingenieros relacionados a la TI. Este universo de legislación y de encuestados fue suficiente para realizar las pruebas de validación planteadas.

6.2 Limitaciones

El sistema de búsqueda funciona si los documentos son correctamente cargados por el administrador. En las pruebas, se tuvo la seguridad de que así sea, para nuevas cargas que aumenten la base de datos, debe cumplirse esta condición.

Los diccionarios implementados de sinónimos y antónimos, así como el de siglas, pueden ser ampliados en su vocabulario, pues se implementaron diccionarios sencillos para el desarrollo inicial, pero fácilmente puede reemplazarse por diccionarios más extensivos. Además, de momento no incluye diccionario de homónimos y parónimos, los cuales aumentarían la capacidad de exhaustividad, obteniendo una mayor cantidad de resultados relevantes.

6.3 Trabajos Futuros

DoLaw fue creado inicialmente como un proyecto para legislación peruana de tecnología de información, pero el universo de legislación puede ser reemplazado por otro sector o ampliado a más sectores.

Además, DoLaw ha probado que se puede y se deben crear sistemas que se puedan usar en la vida real a partir del conocimiento generado en diversos estudios, pues precisamente el conocimiento existe para aplicarlo, y está puesto a disposición para usarlo en diferentes dominios, según las necesidades que se tengan. Esto se lograría reemplazando la base de archivos, y actualizando las características especiales aplicadas para la legislación de TI. Eso significa que muchas de las funcionalidades más importantes de DoLaw, como la interpretación sintáctica y semántica de las consultas de los usuarios, pueden ser utilizadas en otros proyectos que usen un dominio diferente a la legislación.

Otro trabajo futuro puede ser hacer un estudio del porqué hay usuarios que no aceptan el cambio a un nuevo sistema, y trabajar en DoLaw para acercar un poco la herramienta a dichos usuarios, sin perder a los que ya se ha ganado. Por ejemplo, se podría seleccionar los puntos de satisfacción más bajos para trabajar en ellos y, así, lograr mejores resultados de satisfacción. No obstante, hace falta un estudio para comprobar si es la persona la que se resiste al cambio o es la herramienta la que no los convence.

Por último, puede completarse el universo de legislación de TI, e igualarlo al que posea SPIJ, para así poder comparar DoLaw con SPIJ con mayor precisión en nuevas métricas como las del tiempo de respuesta.

REFERENCIAS BIBLIOGRÁFICAS

Armstrong, B., Fogarty, G., Dingsdag, D., & Dimbleby, J. (2005). Validation of a computer user satisfaction questionnaire to measure IS success in small business. *Journal of Research and Practice in Information Technology*, 27-42.

Bast, H., & Weber, I. (2006). Type less, find more: fast autocompletion search with a succinct index. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 364-371.

Berman, J. (2016). Chapter 3 - Indexing Text. In J. Berman, *Data Simplification* (pp. 91-133). Morgan Kaufmann.

Calvillo, A., Padilla, A., & Muñoz, J. (2013). Searching Research Papers Using Clustering and Text Mining. *International Conference on Electronics, Communications and Computing (CONIELECOMP), 2013* (pp. 78-81). Cholula: IEEE.

Chan, F., & Thong, J. (2009). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, 803-814.

Chávez, A., Fernández, A., Dávila, H., Gutiérrez, Y., Collazo, A., & Abreu, J. I. (2013). Textual Similarity based on Lexical-Semantic features. *Second Joint Conference on Lexical and Computational Semantics*, (pp. 109-118). Atlanta, Georgia.

Congreso de la República del Perú. (2016, Junio). *Archivo digital de la legislación del Perú*. Retrieved from Portal del congreso de la república: http://www.leyes.congreso.gob.pe/inicio.aspx

Diario El Peruano. (2019, 03). *El Peruano*. Retrieved from https://diariooficial.elperuano.pe/normas

Ferreyros, C. (2016, Mayo). Entrevista al Dr. Ferreyros, gerente de TI del Ministerio del Trabajo. (D. Otoya, Interviewer)

Guo, K., Zhang, R., & Kuang, L. (2016). TMR: Towards an efficient semantic-based heterogeneous transportation media big data retrieval. *Neurocomputing*, 122-131.

Gurung, D., Chakraborty, U. K., & Sharma, P. (2016). Intelligent Predictive String Search Algorithm. In P. C. 79 (Ed.), *7th International Conference on Communication, Computing and Virtualization 2016* (pp. 161-169). Majhitar, Sikkim, India: Sikkim Manipal Institute of Technology.

Gutiérrez, Y., Vázquez, S., & Montoyo, A. (2016). A semantic framework for textual data enrichment. *Expert Systems With Applications*, Accepted Manuscript.

Hanauer, D., Mei, Q., Law, J., Khanna, R., & Zheng, K. (2015). Supporting information retrieval from electronic health records: A report of University of Michigan's nine-year experience in developing and using the Electronic Medical Record Search Engine (EMERSE). *Journal of Biomedical Informatics*, 290-300.

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering version 2.3.* Durham, UK: Keele University and University of Durham.

Konow, R., & Navarro, G. (2012). Dual-sorted inverted lists in practice. *International Symposium on String Processing and Information Retrieval*, 295-306.

Korteweg, A. (2016, Julio). *Scrum, why we use this agile methodology*. Retrieved from J-Development: http://jdevelopment.nl/scrum-agile-methodology/

Lashkaria, F., Ensanb, F., Bagheric, E., & Ghorbani, A. A. (2017). Efficient indexing for semantic search. *Expert Systems with Applications*, 92-114.

Lei, H., Ganjeizadeh, F., Kumar, P., & Ozcan, P. (2015). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, In Press.

Liddy, E. (2005). Automatic document retrieval. In E. Liddy, *Encyclopedia of Language and Linguistics.*

Liu, C.-T., & Guo, Y. M. (2008). Validating the End-User Computing Satisfaction Instrument for Online Shopping Systems. *Journal of Organizational and End User Computing*, 74-96.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval.* Cambridge University Press.

Martinez, F., & Rodriguez, J. (2004). *Reflexiones sobre la evaluación de los sistemas de recuperación de información: Necesidad, utilidad y viabilidad.* Universidad de Murcia.

Ministerio Público: Fiscalía de la Nación. (2014, Mayo). *Portal del Ministerio Público.* Retrieved from Anuario estadístico 2013: http://portal.mpfn.gob.pe/estadistica/anuario_est_2013.pdf

Ministerio Público: Fiscalía de la Nación. (2015, Abril). *Anuario estadístico 2014.* Retrieved from Portal del Ministerio Público: http://portal.mpfn.gob.pe/estadistica/ANUARIOESTADISTICO2014FINAL.pdf

Mohd, M. (2011). Development of Search Engines using Lucene: An Experience. *Kongres Pengajaran dan Pembelajaran UKM, 2010* (pp. 282-286). Bangi, Malaysia: Procedia Social and Behavioral Sciences.

Moloshnikov, I., Sboev, A., Rybka, R., & Gydovskikh, D. (2015). An algorithm of finding thematically similar documents with creating context-semantic graph based on probabilistic-entropy approach. *4th International Young Scientists Conference on Computational Science* (pp. 297-306). Moscow, Russia: Procedia Computer Science.

Núñez, H., & Ramos, E. (2012). Automatic classification of academic documents using text mining techniques. *XXXVIII Conferencia Latinoamericana en Informatica (CLEI), 2012* (pp. 1-7). Medellín: IEEE.

Radio Capital: Rosa María Palacios. (2011, Febrero 3). *Youtube: qv peru.* Retrieved from Juicio de Alimentos - cuánto cuesta y cuánto demora: https://www.youtube.com/watch?v=jH4svdf1mPo

Rana, N. P., Dwivedi, Y. K., Williams, M. D., & Weerakkody, V. (2015). Investigating success of an e-government initiative: Validation of an integrated IS success model. *Information Systems Frontiers*, 127-142.

Sánchez, D., & Batet, M. (2013). A semantic similarity method based on information content exploiting multiple ontologies. *Expert Systems with Applications*, 1393-1399.

Santisteban, J., & Mauricio, D. (2017). Systematic Literature Review of Critical Success Factors of Information Technology Startups. *Academy of Entrepreneurship Journal*, 1-23.

Schmidt, S., Schnitzer, S., & Rensing, C. (2016). Text classification based filters for a domain-specific search engine. *Computers in Industry*, 70-79.

Scrum Alliance. (2016, Julio). *Who uses Scrum and why?* Retrieved from Scrum Alliance: https://www.scrumalliance.org/why-scrum/who-uses-scrum

The Apache Software Foundation. (2016, Junio). *Apache Lucene Core*. Retrieved from Lucene TM: https://lucene.apache.org/core/

Vargas-Vera, M., Castellanos, T., & Lytras, M. (2010). CONQUIRO: A cluster-based meta-search engine. *Computer in Human Behavior*, 1303-1309.

Zhang, J., Wei, Q., & Chen, G. (2014). A heuristic approach for k-representative information retrieval from large-scale data. *Information Sciences*, 825-841.

ANEXO A

1. Acta de Constitución del proyecto

Acta de Constitución del proyecto

Fecha | hora de la reunión *15/10/2017* | *19:00* | Lugar de la reunión *UNMSM*

Reunión organizada por	Diego Otoya	Diego Otoya
Tipo de reunión	Constitución de proyecto	Carlos Ferreyros
Responsable	Diego Otoya	David Mauricio

TEMAS DE LA AGENDA

Tema de la agenda *Definir el proyecto* | Moderador *Diego Otoya*

Características

Nombre del proyecto:	Desarrollo de DoLaw
Nombre del programa:	DoLaw
Objetivo del proyecto:	Desarrollar un buscador especializado en la búsqueda de documentos en legislación en TI en el periodo 2010 en adelante, a través de la interpretación semántica de las palabras clave que el usuario final introduce, buscando en el contenido completo de dichos documentos, con configuraciones especializadas en legislación de TI
Plazo máximo de entrega:	Diciembre – 2018

Tema de la agenda *Establecer el equipo de trabajo* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Se establece a Diego Otoya como principal desarrollador del proyecto	Diego Otoya	Durante toda la duración del proyecto
Se establece al Dr. David Mauricio como Scrum Manager	David Mauricio	Durante toda la duración del proyecto
Se establece al Dr. Carlos Ferreyros como cliente representante	Carlos Ferreyros	Durante toda la duración del proyecto

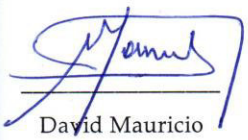
Tema de la agenda *Establecer la problemática* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Se concuerda la problemática:	Explicación de la problemática: Carlos Ferreyros Delimitación de la problemática: Diego Otoya	A detallar en los sprints

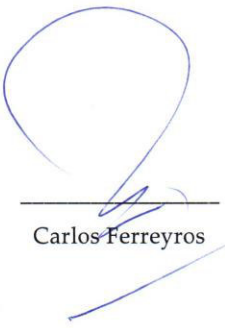
Id	Problemática
P01	Los resultados obtenidos son diferentes si utilizas la misma palabra con o sin tilde
P02	Si se comete un error ortográfico, el buscador no encontrará lo que se quiso buscar
P03	No se puede buscar en varias categorías de leyes a la vez
P04	No se puede descargar la legislación como pdf (En SPIJ) pues la legislación se encuentra incrustada en la web (en HTML)
P05	Falta vista rápida o resumen para comprobar que el documento efectivamente tiene lo buscado en el contexto que se necesitas
P06	La legislación está organizada por fechas y por categorías, pero es posible combinar categorías y/o rangos de fecha
P07	La interfaz gráfica de usuarios es complicada, ya que de entrada presenta demasiadas opciones que un usuario promedio no usa.

Cierre

Los presentes aprueban lo descrito en esta acta de constitución a través de sus firmas



David Mauricio



Carlos Ferreyros

Diego Otoya

2. Acta de Sprint 1

Acta de Reunión de inicial Scrum – Sprint 1

Fecha | hora de la reunión *15/10/2017* | *20:00* | Lugar de la reunión *UNMSM*

Reunión organizada por	Diego Otoya	Diego Otoya
Tipo de reunión	Reunión inicial Scrum	Carlos Ferreyros
Responsable	Diego Otoya	David Mauricio

TEMAS DE LA AGENDA

Tema de la agenda *Creación del Product Backlog inicial* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Se establece el Product Backlog inicial	Establecer, valorar y delimitar las historias de usuario: Diego Otoya Establecer criterios de satisfacción: Carlos Ferreyros Validar el cumplimiento de las historias de usuario en el sistema final: David Mauricio	Durante la duración de la reunión

Id	Descripción	Coste	Condiciones de satisfacción	Valor
HU1	Como cliente, deseo poder visualizar la de manera gráfica como funciona el sistema	600	Diferenciar las características del administrador con las del usuario, en un diagrama de arquitectura	700
HU2	Como cliente, deseo que la legislación sea almacenada en pdf	800	Debe mantenerse los archivos en formato pdf	600
HU3	Como cliente, deseo que el producto tenga un control de acceso (Login) en caso deba ingresar como administrador	400	El login debe consistir de id y contraseña y dará accesos de administrador. Un usuario regular no necesitará login	300
HU4	Como administrador, deseo registrar nueva legislación al sistema	500	Los documentos de legislación deben poder ingresarse mediante el sistema	800
HU5	Como administrador, deseo actualizar la legislación ingresada	600	Los documentos de legislación deben poder actualizarse cuando se desee	700
HU6	Como administrador, deseo un diseño de interfaz gráfica agradable, para realizar mantenimiento	300	Debe aprobarse el diseño realizado antes de realizar la implementación	800
HU7	Como usuario, deseo un diseño de interfaz de búsqueda fácil de usar	400	Debe aprobarse el diseño realizado antes de realizar la implementación	900

Medidas					Persona responsable	Fecha límite
HU8	Como usuario, deseo que el sistema busque la documentación que coincida con las palabras clave ingresadas, y con sus sinonimos	1000	Debe permitirse palabras o frases clave para realizar la búsqueda, y que la búsqueda se haga por sus sinónimos	1000		
HU9	Como usuario, deseo recibir con precisión los resultados obtenidos	900	Debe haber un tratamiento especial de las consultas para mostrar la mayor cantidad de resultados relevantes sobre el total de resultados	900		
HU10	Como usuario, deseo filtrar la legislación por categorías o fecha de publicación	900	Se debe ofrecer el filtrado por categoría y fecha	800		
HU11	Como usuario, deseo poder descargar la legislación correspondiente a los resultados obtenidos o elegidos	200	La descarga debe hacerse en pdf	500		
HU12	Como usuario, deseo que los resultados de mi búsqueda esten ordenados por prioridad	300	La prioridad será definida colocando primero el documento con mayor relevancia con la consulta	300		
HU13	Como usuario, deseo que el sistema contenga siglas y abreviaciones correspondientes al area legislativa	800	La busqueda debe poder realizarse tanto buscando las siglas o abreviación, como buscando la palabra completa	800		
HU14	Como usuario, deseo que al ingresar una consulta, se ignore sufijos de género y número en las palabras ingresadas	800	El sistema debe obtener las raíces de las palabras ingresadas y realizar la búsqueda mediante estas	900		
HU15	Como usuario, deseo que los resultados se obtengan en un tiempo de respuesta menos a 2 segundos	600	El tiempo de respuesta del servidor será de dos segundos. Además, se plantea una interfaz que reduzca el tiempo de carga de la web desde internet	500		

Tema de la agenda *Definir el Sprint inicial* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Se escogen las historias de usuario que pertenecen al primer sprint	Diego Otoya David Mauricio	Durante la reunión
Desarrollar las historias de usuario elegidas:	Diego Otoya	15-11-2017

Id	Descripción
HU1	Como cliente, deseo poder visualizar de manera gráfica cómo funciona el sistema
HU2	Como cliente, deseo que la legislación sea almacenada en pdf
HU3	Como cliente, deseo que el producto tenga un control de acceso (Login) en caso deba ingresar como administrador

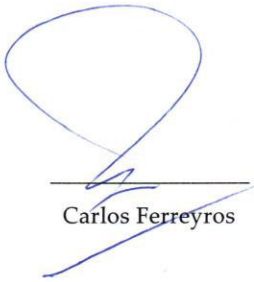
Ajustar los criterios de cumplimiento para las historias de usuario	Carlos Ferreyros	Durante la reunión

Id	Condiciones de satisfacción
HU1	Diferenciar las características del administrador con las del usuario, en un diagrama del sistema
HU2	La legislación debe almacenarse en un servidor de archivos en formato pdf
HU3	El login debe consistir de id y contraseña y dará accesos de administrador. Un usuario regular no necesitará login

Cierre

Los presentes aprueban lo descrito en esta acta a través de sus firmas

David Mauricio



Carlos Ferreyros

Diego Otoya

3. Acta de Sprint 2

Acta de Sprint 2

Fecha | hora de la reunión *15/11/2017* | *20:00* | Lugar de la reunión *UNMSM*

Reunión organizada por	Diego Otoya	Diego Otoya
Tipo de reunión	Reunión Sprint 2	Carlos Ferreyros
Responsable	Diego Otoya	David Mauricio

TEMAS DE LA AGENDA

Tema de la agenda *Aprobar el Sprint 1* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Revisar que las historias del Sprint 1 se hayan terminado satisfactoriamente	David Mauricio Carlos Ferreyros	Durante la reunión
No hubo observaciones		

Tema de la agenda *Definir el Sprint 2* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Se escogen las historias de usuario que pertenecen al segundo sprint	Diego Otoya David Mauricio	Durante la reunión
Desarrollar las historias de usuario elegidas:	Diego Otoya	15-01-2018

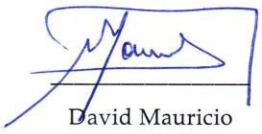
Id	Descripción
HU4	Como administrador, deseo registrar nueva legislación al sistema
HU5	Como administrador, deseo actualizar la legislación ingresada
HU6	Como administrador, deseo un diseño de interfaz gráfica agradable, para realizar mantenimiento

Medidas	Persona responsable	Fecha límite
Ajustar los criterios de cumplimiento para las historias de usuario	Carlos Ferreyros	Durante la reunión

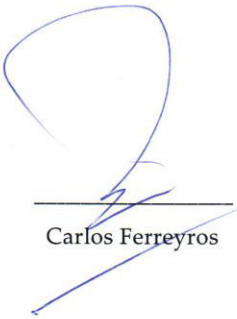
Id	Condiciones de satisfacción
HU4	Los documentos de legislación deben poder ingresarse mediante el sistema
HU5	Los documentos de legislación (pdf) deben poder actualizarse cuando se desee
HU6 *	Debe aprobarse el maquetado realizado antes de realizar la implementación del diseño

Cierre

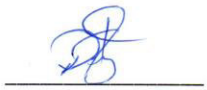
Los presentes aprueban lo descrito en esta acta a través de sus firmas



David Mauricio



Carlos Ferreyros



Diego Otoya

4. Acta de Sprint 3

Acta de Sprint 3

Fecha | hora de la reunión *15/01/2018* | *20:00* | Lugar de la reunión *UNMSM*

Reunión organizada por	Diego Otoya	Diego Otoya
Tipo de reunión	Reunión Sprint 3	Carlos Ferreyros
Responsable	Diego Otoya	David Mauricio

TEMAS DE LA AGENDA

Tema de la agenda *Aprobar el Sprint 2* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Revisar que las historias del Sprint 2 se hayan terminado satisfactoriamente Se dieron sugerencias en el maquetado inicial del proyecto. La historia HU6 se corregirá en esta iteración	David Mauricio Carlos Ferreyros	Durante la reunión

Tema de la agenda *Definir el Sprint 3* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Se escogen las historias de usuario que pertenecen al tercer sprint	Diego Otoya David Mauricio	Durante la reunión
Desarrollar las historias de usuario elegidas:	Diego Otoya	15-03-2018

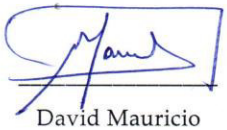
Id	Descripción
HU6 *	Como administrador, deseo un diseño de interfaz gráfica agradable, para realizar mantenimiento
HU7	Como usuario, deseo un diseño de interfaz de búsqueda fácil de usar
HU8	Como usuario, deseo que el sistema busque la documentación que coincida con las palabras clave ingresadas, y con sus sinónimos

Medidas	Persona responsable	Fecha límite
Ajustar los criterios de cumplimiento para las historias de usuario	Carlos Ferreyros	Durante la reunión

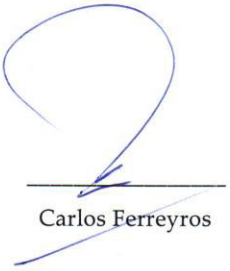
Id	Condiciones de satisfacción
HU6 *	Debe aprobarse el maquetado realizado antes de realizar la implementación del diseño
HU7	Debe aprobarse el diseño realizado antes de realizar la implementación
HU8	Debe permitirse palabras o frases clave para realizar la búsqueda, y que la búsqueda se haga por sus sinónimos

Cierre

Los presentes aprueban lo descrito en esta acta a través de sus firmas



David Mauricio



Carlos Ferreyros

Diego Otoya

5. Acta de Sprint 4

Acta de Sprint 4

Fecha | hora de la reunión *23/03/2018* | *20:00* | Lugar de la reunión *UNMSM*

Reunión organizada por	Diego Otoya	Diego Otoya
Tipo de reunión	Reunión Sprint 4	Carlos Ferreyros
Responsable	Diego Otoya	David Mauricio

TEMAS DE LA AGENDA

Tema de la agenda *Aprobar el Sprint 3* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Revisar que las historias del Sprint 3 se hayan terminado satisfactoriamente	David Mauricio Carlos Ferreyros	Durante la reunión
Se levantaron las observaciones a la historia HU6		
No hubo nuevas observaciones		

Tema de la agenda *Definir el Sprint 4* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Se escogen las historias de usuario que pertenecen al cuarto sprint	Diego Otoya David Mauricio	Durante la reunión
Desarrollar las historias de usuario elegidas:	Diego Otoya	15-06-2018

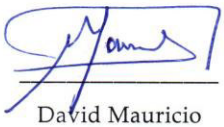
Id	Descripción
HU9	Como usuario, deseo recibir con precisión los resultados obtenidos
HU10	Como usuario, deseo filtrar la legislación por categorías o fecha de publicación
HU11	Como usuario, deseo poder descargar la legislación correspondiente a los resultados obtenidos o elegidos
HU12	Como usuario, deseo que los resultados de mi búsqueda estén ordenados por prioridad

Medidas	Persona responsable	Fecha límite
Ajustar los criterios de cumplimiento para las historias de usuario	Carlos Ferreyros	Durante la reunión

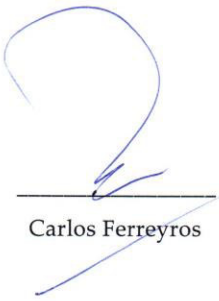
Id	Condiciones de satisfacción
HU9	Debe haber un tratamiento especial de las consultas para mostrar la mayor cantidad de resultados relevantes sobre el total de resultados
HU10	Se debe ofrecer el filtrado por categoría y fecha
HU11 *	La descarga de los resultados debe hacerse en pdf, además de la vista previa
HU12	La prioridad será definida colocando primero el documento con mayor relevancia con la consulta

Cierre

Los presentes aprueban lo descrito en esta acta a través de sus firmas



David Mauricio



Carlos Ferreyros

Diego Otoya

6. Acta de Sprint 5

Acta de Sprint 5

Fecha | hora de la reunión *22/06/2018* | *20:00* | Lugar de la reunión *UNMSM*

Reunión organizada por	Diego Otoya	Diego Otoya
Tipo de reunión	Reunión Sprint 5	Carlos Ferreyros
Responsable	Diego Otoya	David Mauricio

TEMAS DE LA AGENDA

Tema de la agenda *Aprobar el Sprint 4* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Revisar que las historias del Sprint 4 se hayan terminado satisfactoriamente No hubo observaciones	David Mauricio Carlos Ferreyros	Durante la reunión

Tema de la agenda *Definir el Sprint 4* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Se escogen las historias de usuario que pertenecen al cuarto sprint	Diego Otoya David Mauricio	Durante la reunión
Aprovechar la oportunidad de incluir la búsqueda semántica, que consiste en buscar no solo lo que escribe el usuario, sino lo que quiso decir. Dicha característica se la agregó en la HU16	Diego Otoya	15-10-2018
Desarrollar las historias de usuario elegidas:	Diego Otoya	15-10-2018

Id	Descripción
HU13	Como usuario, deseo que el sistema contenga siglas y abreviaciones correspondientes al area legislativa
HU14	Como usuario, deseo que al ingresar una consulta, se ignore sufijos de género y número en las palabras ingresadas

Medidas		Persona responsable	Fecha límite
HU15	Como usuario, deseo que los resultados se obtengan en un tiempo de respuesta menos a 2 segundos		
HU16 *	Como usuario, deseo que el sistema entienda lo que estoy buscando (búsqueda semántica), incluso si cometo un error ortográfico		

Ajustar los criterios de cumplimiento para las historias de usuario	Carlos Ferreyros	Durante la reunión

Id	Condiciones de satisfacción
HU13	La búsqueda debe poder realizarse tanto buscando las siglas o abreviación, como buscando la palabra completa
HU14	El sistema debe obtener las raíces de las palabras ingresadas y realizar la búsqueda mediante estas
HU15	El tiempo de respuesta del servidor será de dos segundos. Además, se plantea una interfaz que reduzca el tiempo de carga de la web desde internet
HU16	El sistema debe buscar palabras similares a las ingresadas; y debe hacer un tratamiento de la consulta para realizar subconsultas que el usuario pueda necesitar a partir de lo ingresado

Cierre

Los presentes aprueban lo descrito en esta acta a través de sus firmas

David Mauricio

Carlos Ferreyros

Diego Otoya

7. Acta de Cierre

Acta de Cierre

Fecha | hora de la reunión *15/10/2018* | *18:00* | Lugar de la reunión *UNMSM*

Reunión organizada por	Diego Otoya	Diego Otoya
Tipo de reunión	Reunión de Cierre de Proyecto	Carlos Ferreyros
		David Mauricio
Responsable	Diego Otoya	

TEMAS DE LA AGENDA

Tema de la agenda *Aprobar el Sprint 5* | Moderador *Diego Otoya*

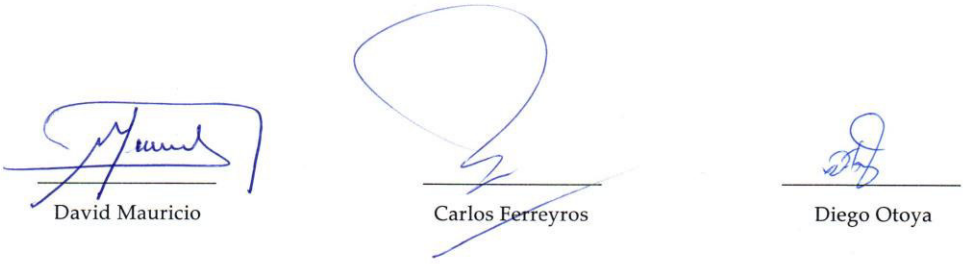
Medidas	Persona responsable	Fecha límite
Revisar que las historias del Sprint 5 se hayan terminado satisfactoriamente	David Mauricio Carlos Ferreyros	Durante la reunión
Se sugirió aumentar las palabras de los diccionarios del sistema, pero se aprobó el proyecto		

Tema de la agenda *Presentar una demo del proyecto* | Moderador *Diego Otoya*

Medidas	Persona responsable	Fecha límite
Realizar la presentación del proyecto y dar el proyecto por terminado	Diego Otoya	15-11-2018

Cierre

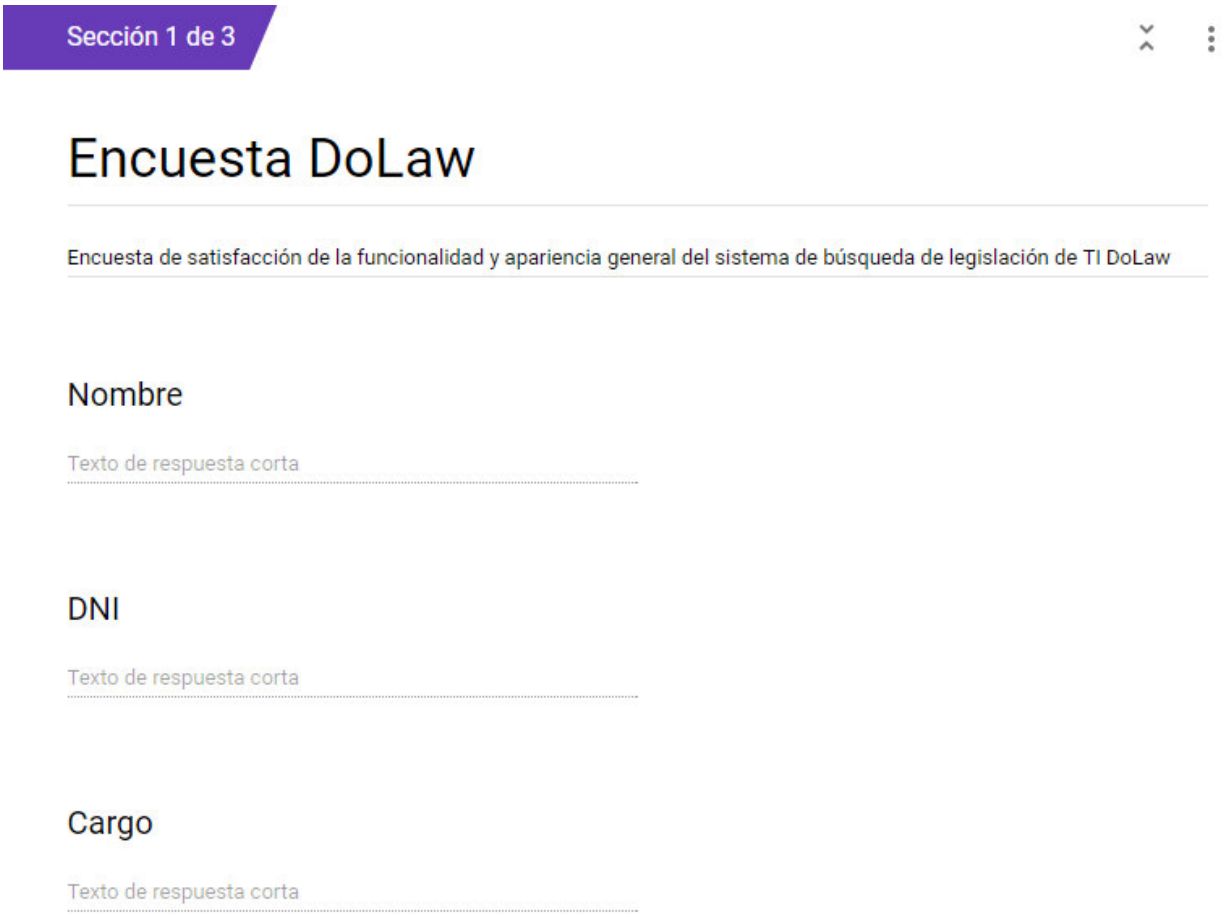
Los presentes aprueban lo descrito en esta acta a través de sus firmas



David Mauricio | Carlos Ferreyros | Diego Otoya

ANEXO B

1. Encuesta DoLaw



Sección 2 de 3

Funcionalidad

Descripción (opcional)

¿Cree ud. que los filtros ofrecidos por DoLaw mejores que los ofrecidos por SPIJ?

	1	2	3	4	5	
Mucho peores	○	○	○	○	○	Mucho mejores

¿Cree ud. que la velocidad de obtención de resultados por DoLaw son mejores que los ofrecidos por SPIJ?

	1	2	3	4	5	
Mucho peores	○	○	○	○	○	Mucho mejores

¿Cual cree ud. que es la precisión de los resultados obtenidos por SPIJ con respecto a su cadena de búsqueda?

	1	2	3	4	5	
Muy mala	○	○	○	○	○	Muy buena

¿Cual cree ud. que es la precisión de los resultados obtenidos por DoLaw con respecto a su cadena de búsqueda?

	1	2	3	4	5	
Muy mala	○	○	○	○	○	Muy buena

¿Que tan satisfecho lo dejaría un sistema con las funcionalidades de DoLaw?

	1	2	3	4	5	
Muy insatisfecho	○	○	○	○	○	Muy satisfecho

Sección 3 de 3

Apariencia

Descripción (opcional)

¿Que opina de la apariencia general del sistema SPIJ?

	1	2	3	4	5	
Muy mala	○	○	○	○	○	Muy buena

¿Que opina de la apariencia general del sistema DoLaw?

	1	2	3	4	5	
Muy mala	○	○	○	○	○	Muy buena

¿Cuan fácil de usar es SPIJ?

	1	2	3	4	5	
Muy difícil	○	○	○	○	○	Muy fácil

¿Cuan fácil de usar es DoLaw?

	1	2	3	4	5	
Muy difícil	○	○	○	○	○	Muy fácil

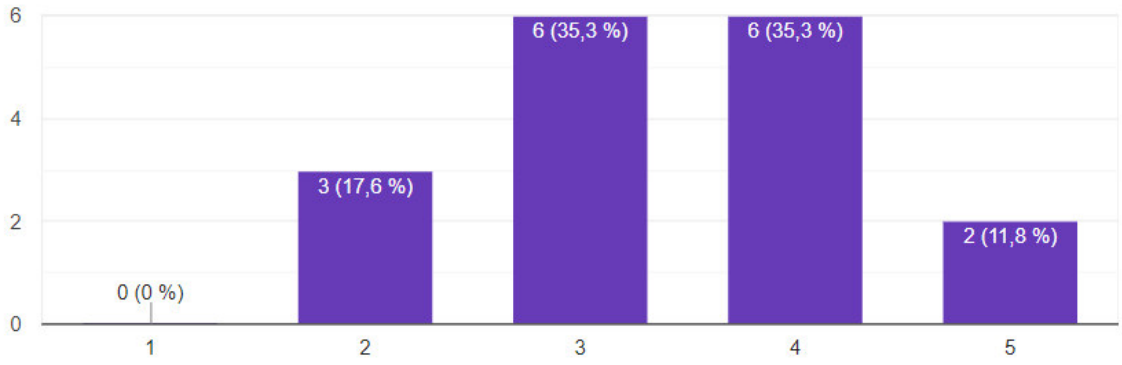
En caso de que ambos sistemas tengan el mismo contenido de leyes, ¿Cuál sistema preferiría usar?

- ○ SPIJ
- ○ DoLaw

2. Respuestas dadas en la encuesta

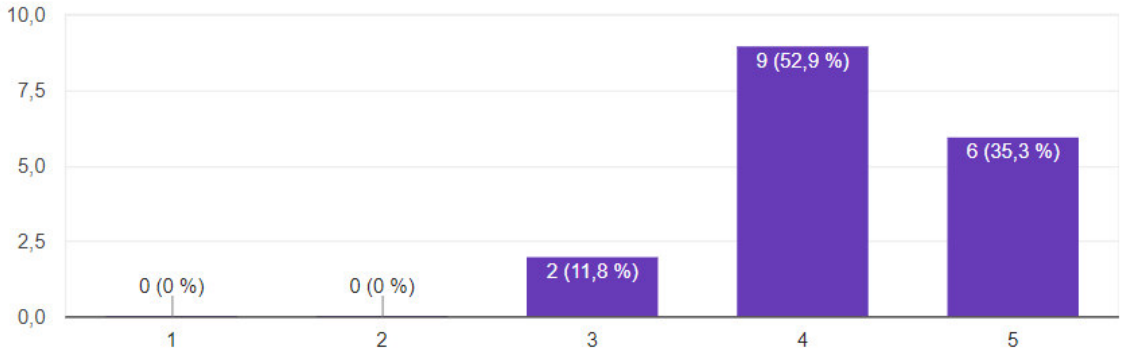
¿Cree ud. que los filtros ofrecidos por DoLaw mejores que los ofrecidos por SPIJ?

17 respuestas



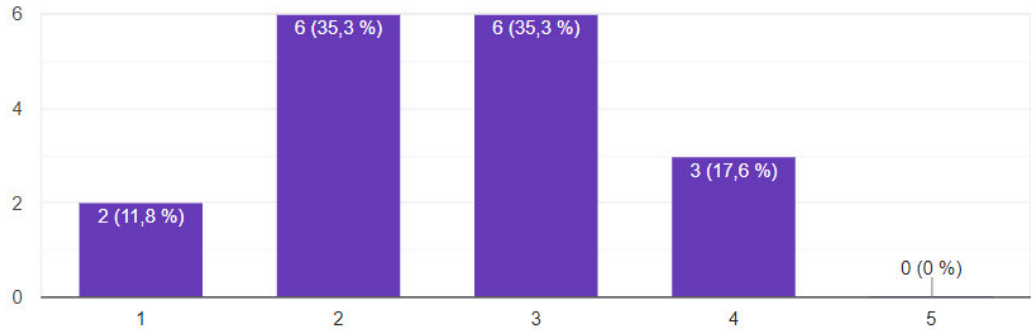
¿Cree ud. que la velocidad de obtención de resultados por DoLaw son mejores que los ofrecidos por SPIJ?

17 respuestas



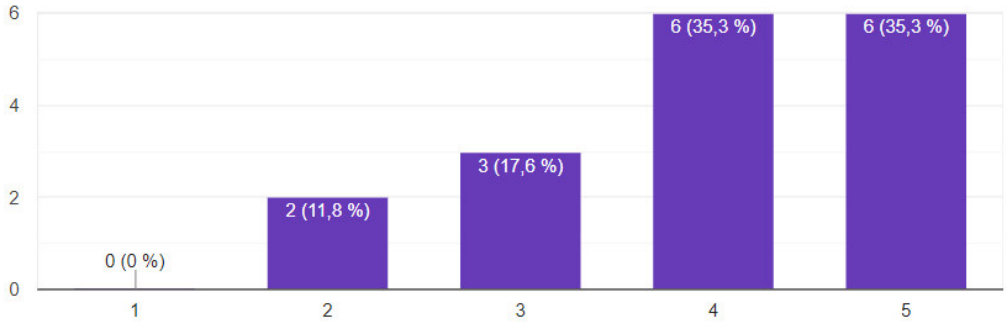
¿Cual cree ud. que es la precisión de los resultados obtenidos por SPIJ con respecto a su cadena de búsqueda?

17 respuestas



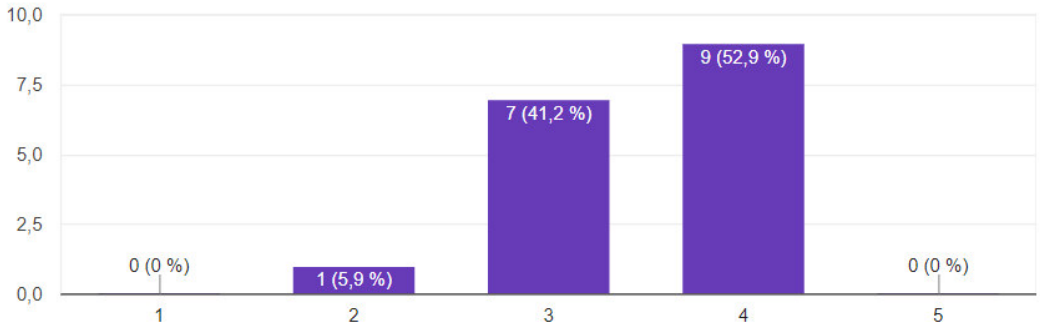
¿Cual cree ud. que es la precisión de los resultados obtenidos por DoLaw con respecto a su cadena de búsqueda?

17 respuestas



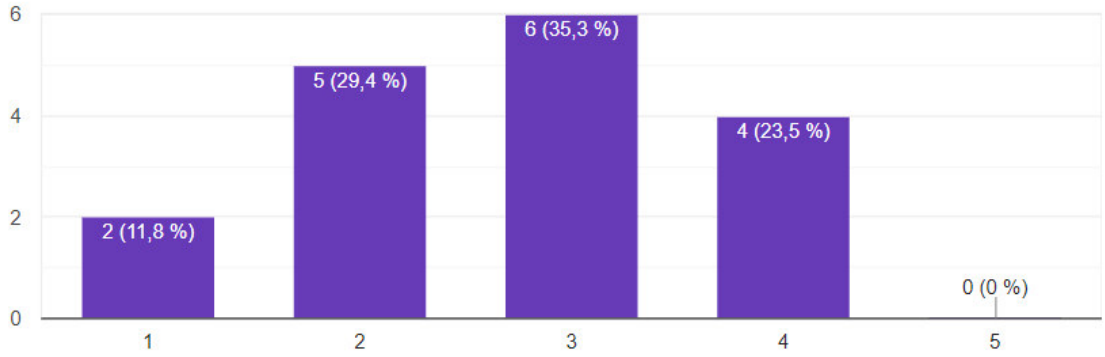
¿Que tan satisfecho lo dejaría un sistema con las funcionalidades de DoLaw?

17 respuestas



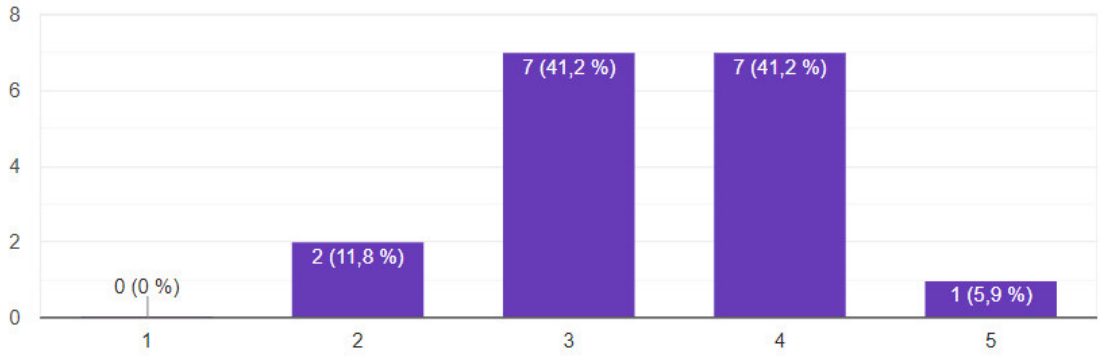
¿Que opina de la apariencia general del sistema SPIJ?

17 respuestas



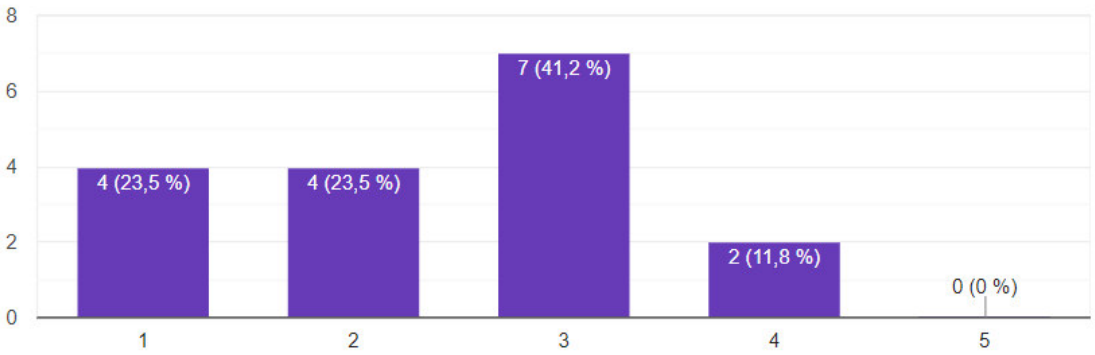
¿Que opina de la apariencia general del sistema DoLaw?

17 respuestas



¿Cuan fácil de usar es SPIJ?

17 respuestas



¿Cuan fácil de usar es DoLaw?

17 respuestas